

** DE 6502 KENNERS ** -- EEN CLUB VOOR 65xx GEBRUIKERS

De vereniging heeft leden in Nederland, België, Duitsland, Frankrijk, Spanje, Portugal, Amerika, India. Het doel van de vereniging is: het bevorderen van de kennisuitwisseling tussen de gebruikers van 65xx-computers, zoals KIM, JUNIOR, COMMODORE-64, APPLE, CHE-1, PEARCOM, AIM-65, SYM, PET, BBC, ATARI, VIC-20, BASIS 108, PROTON COMPUTERS, ITT 2020, OSI, ACC 8000, ACORN ELECTRON, SYSTEM 65, PC-100, PALLAS, MINTA FORMOSA, ORIC-1, STARLIGHT, CV-777, ESTATE III, SBC 65/68, NCS 6502, KEMPAC System 4, Elektuur SAMSON-65 DOS computer. De kennis uitwisseling wordt o.a. gerealiseerd door 5 maal per jaar DE 6502 KENNER te publiceren (1984 en 1985 6 maal zonder contributieverhoging), door het houden van clubbijeenkomsten, door een cassette-bibliotheek in stand te houden en door het verlenen van papierware-service. De leden zelf organiseren regio-bijeenkomsten.

Verschijningsdata DE 6502 KENNER

derde zaterdag
van de maanden:
februari, april, juni,
augustus, oktober, december.

Inlichtingen over de regio- bijeenkomsten:

Gerard van Roekel,
Van der Palmstraat 11 - C
3135 LK Vlaardingen.
Tel.: 010 - 351101

De vereniging is volledig onafhankelijk, is statutair oogericht en ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar, onder nummer 634305.

Voorzitter:
Rinus Vleesch-Dubois
Fl. Nightingalestraat 212
2037 NG Haarlem
Tel.: 023 - 330993

Penningmeester:
John F. van Sorang
Tulp 71
2925 EW Krimpen/IJssel.
Tel.: 01807 - 20589

Leden:

| | | |
|-------------------|--|-------------------|
| Adri Hankel | (05490 - 51151) | Hardware/software |
| Jan Tomoot | (070 - 931417) | Promotie |
| Erwin Visschedijk | (05490 - 71416) | Hardware/software |
| Nico de Vries | (010 - 502239) | |
| Erevoorzitter: | Siep de Vries | |
| Ereleden : | Mw. H. de Vries - Van der Winden | |
| | Anton Mueller | |
| Lidmaatschap : | Fl. 45,== per kalenderjaar, postrekening 3757649 t.n.v. KIM Gebruikers Club Ned. te Krimpen a.d. IJssel. | |
| Advertenties : | Tarieven op aanvraag bij de redactie. | |

Bijeenkomsten van de club

derde zaterdag
van de maanden:
januari, maart, mei,
september, november.

Redactie-adres en informatie over papierware etc.:

Willem L. van Pelt
Jacob Jordaensstraat 15
2923 CK Krimpen/IJssel.
Tel.: 01807 - 19881

Sekretaris:
Gert Klein
Diedenweg 119
6706 CM Wageningen
Tel.: 08370 - 23646

Redactie DE 6502 KENNER:
Willem L. van Pelt
Jacob Jordaensstraat 15
2923 CK Krimpen/IJssel.
Tel.: 01807 - 19881

** DE 6502 KENNER ** -- EEN BLAD VOOR 65xx GEBRUIKERS

DE 6502 KENNER is een uitgave van de KIM Gebruikers Club Nederland. Het blad wordt verstrekt aan leden van de club. DE 6502 KENNER wordt van copy voorzien door leden van de club, bij de oomak van een publikatie bijgestaan door de redactie. De inzendingen van programma's dienen voorzien te zijn van commentaar in de listings. Dit, in verband met het educatieve aspect. De inzendingen kunnen geschieden in machinetaal, de assembler/disassembler/editor Micro-ADE, de Format Lister/Assembler/Tape-utilities/Editor FATE, de resident assembler/editor van Carl Moser, in de hogere programmeertalen zoals bijvoorbeeld Basic, Basicode, Forth, Focal, Comal, Pascal, etc. etc. De leden schrijven ook artikelen over de door hen ontwikkelde hardware en/of aanpassingen daarop. Zij schrijven tevens artikelen van algemene aard of reageren op gepubliceerde artikelen.

DE 6502 KENNER is een blad van de leden en door de leden.

De Micro-ADE is een produkt van Micro-Ware Ltd., geschreven door Peter Jennins en bestemd voor alle 6502-computers. De Kim Gebruikers Club Nederland heeft de copyrights verworven nadat Sebo Woldringh de oorspronkelijke 4K KIM-1 versie met 4K uitbreidde tot 8K KIM-1 versie, welke later door Adri Hankel werd aangepast voor de JUNIOR. Uit de berg aanvullingen en wijzigingen stelde Willem van Pelt een nieuwe 8K source-listing voor de JUNIOR samen. De implementatie op andere systemen dan de KIM-1 en de JUNIOR kan eenvoudig gebeuren door het aanpassen van de I/O-adressen, welke in de source-listing gemakkelijk terug te vinden zijn. Vraag de prijslijst bij de redactie door het zenden van een gefrankeerde (fl.1.40) retourenvelope.

De Format Lister/Assembler/Tape-utilities/Editor FATE staat dankzij de medewerking van Proton Electronics te Naarden nu voor de bezitters van een JUNIOR-computer ter beschikking. Het 12K programma vindt haar oorsprong in het universele disk operating system van Proton Electronics. Rob Banen publiceert daarover enige wijzigingen en schreef daarvoor FATE. Eenmaal met FATE bekend, dan is de stap naar het door ons gewaardeerde disk operating system nog slechts een kwestie van geld. FATE wordt beschikbaar gesteld met toestemming van Proton Electronics te Naarden.

In de edities van DE 6502 KENNER worden regelmatig aankondigingen gedaan van de door de club georganiseerde landelijke en regionale bijeenkomsten. Ook worden bestuurlijke mededelingen gedaan, naast informatie over hetgeen op de markt te koop is. Leden kunnen het gevraagd of te koop aangeboden artikel bekend maken of brieven aan de redactie laten publiceren.

Elke editie van DE 6502 KENNER omvat minimaal 48 pagina's.

De redactie verwacht inzendingen van gebruikers van alle 6502 systemen. Het publiceren van artikelen wordt op de voet gevolgd door een onafhankelijke jury van 6502 kenners bij uitstek. Deze jury kent jaarlijks een aantal aanmoedigingspremies toe voor de beste publikatie in DE 6502 KENNER.

De 6502 KENNER is een uitgave van de KIM gebruikers Club Nederland.

Adres voor het inzenden van en reacties op artikelen voor DE 6502 KENNER:
Willem L. van Pelt
Jacob Jordaensstraat 15
2923 CK Krimpen a/IJssel
Tel.: 01807 - 19881

Vaste medewerkers:
Willem L. van Pelt
Gerard van Roekel
Frans Smeehuijzen
Jaap van Toledo

Freelance medewerkers:
Frans Bakx
Rob Banen
Fridus Jonkman
Gert Klein
Roger Langeveld
Anton Mueller
Gert van Oobroek
Ruud Uhoff

Illustraties:
Jack Uchtman

Gehele of gedeeltelijke overname van de inhoud van DE 6502 KENNER zonder toestemming van het bestuur is verboden. Toepassing van gepubliceerde programma's, hardware etc. is alleen toegestaan voor persoonlijk gebruik.

DE 6502 KENNER verschijnt 6 x per jaar en heeft een oplage van 500 exemplaren.

Copyright (C) 1985 KIM Gebruikers Club Nederland.

De voorpagina is een aquarel van een KIM, geschilderd door:
Rinus Vleesch Dubois.

In verband met auteurswetgeving en andere maatregelen op het gebied van bescherming van software kan de redactie geen aansprakelijkheid aanvaarden voor inzendingen. Inzendingen dienen afkomstig te zijn van de inzender, tenzij anders aangegeven.

INHOUDSOPGAVE DE 6502 KENNER NR. 37 MEI 1985

| | |
|---|---|
| 1. Van de redactie | 2. |
| 2. Tips voor Commodore 64 | 2. |
| 3. Tips & Tricks voor Commodore 64 ... Gerard van Roekel | 3. |
| 4. Uitnodiging bijeenkomst Almelo. mei 1985 | 4. |
| 5. JUNIOR met VDU-kaart GREEDY. deel 3 (slot) ... Phons Bloemen | 5. |
| 6. JUNIOR met DOSAV2.0A Bug in Disc patches on ASSM/TED ... Hans Buurman | 18. |
| 7. JUNIOR met Proton Dos Proton Utility Source ... Piet Fransen | 19. |
| 8. JUNIOR met OHIO Dos Aanpassing OS65D V3.3 PRINT&(X,Y)-commando ... Jan van Heuven | 25. |
| DATBAS. een bestandsprogramma ... Jan van Heuven | 27. |
| 9. COMMODORE 64 Centronics Printer Interface ... Ruud Uhoff | 37. |
| 10. BASIC Tokenized Microsoft Basic Keywords and Adresses ... Nico de Vries | 43. |
| 11. FORTH FORTH op JUNIOR-computer. deel 2 ... Gert van Oobroek | 45. |
| FORTH Extensions ... Fridus Jonkman | 47. |
| 12. DIVERSEN Geheugenplaats bekijken/wijzigen met Basic Waarschuwing over resetschakelaar Commodore 64 Plotterpennen Vraag en Aanbod Brief aan de redactie Boekbespreking Blokken ASCII-keyboard. A.C. Tijmons | 17. 26. 2.36. 42. 42. 48. 49. |

Op de landelijke bijeenkomst te Geldrop werd door onze bestuursleden A. Hankel en E. Visschedijk het systeem geïntroduceerd waarmee zij beiden in middels, en blijkbaar met plezier, werken: CPU kaart, VDUkaart, uitgebreid ramgeheugen, eigen BK monitor en het diskoperating systeem DOS65 van ons clublid Ad Brouwer.

Aan deze computer, die draait op de Elekturbus, is geen eigen naam verbonden. Men heeft de neiging te spreken over de JUNIOR met het DOS65. Dat lijkt me echter niet juist, omdat hier het JUNIOR concept geheel verlaten is. Memorymap zowel als hardware en software hebben nauwelijks iets te maken met JUNIOR.

Het getoonde nieuwe concept heeft de aanwezigen blijkbaar zo geboeid dat ons uit Frankrijk overgekomen lid Drs. Ch. Kleipool, inmiddels ook werkend met het systeem, zich liet ontvallen: "nu de JUNIOR dood is ...". Deze opmerking heeft mij aan het denken gezet. Ik doe het nu even hardop, aannemend dat m'n vriend Kleipool de waarde van mijn opmerkingen weet in te schatten.

In de eerste plaats lijkt het mij goed even terug te blikken naar de beginperiode van JUNIOR. Er is wat voor te zeggen dat de JUNIOR tot stand kwam onder druk van de tijd: eigenlijk te hoge componentkosten voor een redelijk computerconcept voor zelfbouwers. De concessies aan dit punt zijn niet onopgemerkt gebleven. Maar de leden van onze club wisten er wel raad mee. In de afgelopen paar jaar is toch heel goed zichtbaar geworden wat er met JUNIOR te doen valt, al moet gezegd dat soms het creatief vermogen wel heel erg op de proef werd gesteld. Sommigen hebben het gevoel dat men niet in staat bleek de JUNIOR een uniform jasje aan te trekken. En ze lijken gelijk te hebben als we aan het gebruik van verschillende Basics denken. Toch heb ik eerder de neiging te zeggen dat de JUNIOR juist laat zien welke verschillende mogelijkheden er zijn. Wie dat wil kan met een VDU-kaart werken. De JUNIOR kan werken met het Proton-Dos, met OHIO-Dos OS65D V3.3 en het Dos van Koen van Nieuwenhove uit België.

Nu Elektuur in Duitsland al successen bereikt met haar SAMSON65 Dos-computer, is het niet zo heel verwonderlijk dat uit de geleerden iets ooklinkt als "JUNIOR is dood". Het is echter feitelijk onjuist. Het is in het land der prijzen nog steeds niet zo dat de hobbyist koning is. Draaien op het floppy-nivo is stomweg niet voor iedere beurs bereikbaar. En dan ben je aangewezen op cassettes. Kant-en-klaar machines bieden niet altijd datgene wat met zelfbouw-projecten wel bereikbaar blijkt: zeer stevige parate kennis over wat de combinatie software en hardware aankan. Zelfbouw betekent niet dat er geen problemen meer zijn, maar met de kant-en-klaar computer, ik denk bijvoorbeeld aan de Commodore 64 en met name aan het trage disk operating system, blijft er ook nog het nodige te wensen over. Hoe je ook went of keert, naar mijn oordeel zullen er altijd mensen blijven die het gewoon boeiend blijven vinden zelf te bouwen. En het is nu juist hier dat de JUNIOR een uitkomst is. JUNIOR is derhalve niet dood. Integendeel!

W.L. van Pelt.

C-64 TIP 1 M. v.d. Velde, Helmond

Het verbinden van programma's met de Commodore 64.

De te verbinden programma's mogen NIET dezelfde regelnummers hebben.

1. LOAD Laadt het eerste programma.
2. PRINT PEEK (43) Noteer dit getal.
3. PRINT PEEK (44) Noteer dit getal.
4. POKE 43, (PEEK(45)+PEEK(46)*256-2)AND255
5. POKE 44, INT((PEEK(45)+PEEK(46)*256-2)/256)
6. LOAD Laadt het tweede programma.
7. POKE 43,x x is het getal uit punt 2.
8. POKE 44,x x is het getal uit punt 3.
9. LIST U ziet nu een listing van de twee programma's.
10. De punten 2 t/m 8 kunnen herhaald worden indien nog meer programma's verbonden moeten worden.

C-64 TIP 2 M. v.d. Velde, Helmond.

De adressen 2049 en 2050.

Deze twee adressen hebben invloed op/of worden beïnvloed door de Basic-commando's LIST en NEW.

1. Maak een programma van enkele regels.
2. PRINT PEEK(2049),PEEK(2050)
Noteer de 2 getallen.
3. LIST U ziet nu programma.
4. POKE 2049,0 U ziet nu alleen de eerste regel.
5. POKE 2050,0 U krijgt geen listing meer.
6. POKE 2049 en 2050 Met de getallen uit punt 2. Weer een volledige listing.
7. Het commando NEW wist Uw programma (schijnbaar) uit het geheugen.
8. POKE 2049 en 2050 Met de getallen uit punt 2.
9. Uw programma is weer beschikbaar.
Geef maar LIST.

VIDITEL

I would like to communicate with Dutch Viditel from my home in the south of France.
Who can provide me with 6502-Junior software for the Dutch Viditel system?

Or better still: who has the courage to write new or adapt existing software and publish this in DE 6502 KENNER?

Our editor Willem van Pelt told me that he would very much welcome publications on this matter.

Coen Kleipool, Val de Perier, F-83310 Cogolin, France.
T.(94) 54.43.82

T I P S & T R I C K S

SPECIAAL VOOR DE COMMODORE 64 BEZITTERS.

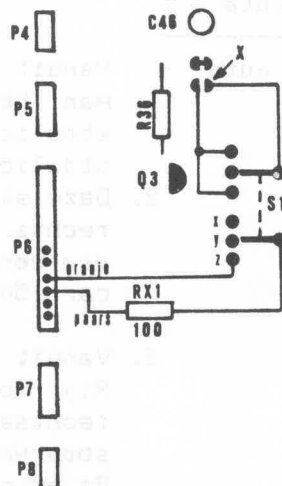
Er zijn nog steeds veel mensen die het gebruik van diskettes aan twee zijden in de 1541 disk drive afraden. Toch wordt dit veelvuldig (en met veel succes) toegepast. In de meeste gevallen wordt een keepje aan de andere zijde van de diskette gemaakt. Hiervoor zijn speciale mesjes in de handel. Het kan ook goedkoper en zonder een keepje in Uw diskette te maken met het hieronder beschreven schema.

Benodigheden:

Dubbelpolige schakelaar met middenstand
Weerstand 100 ohm 1/8 watt

Schema verklaring:

In het schema is een gedeelte van de printplaat getekend, zoals deze in de 1541 disk drive voorkomt. Op de konnektor P6 bevinden zich naast elkaar een oranje en een paarse draad; deze twee gaan via een weerstand van 100 ohm naar een schakelaar. In de maakstand van deze schakelaar is de 'write protect' van de drive opgeheven. De andere zijde van de schakelaar wordt gebruikt voor het schakelen van de drive van device 8 naar device 9. Let wel op dat de doorverbinding, gemerkt X doorsneden moet worden. Als U gebruik wenst te maken van device 9 en write protect off kan dat door eerst de drive op de Y-stand te initialiseren om daarna over te schakelen naar stand Z.



Schakelstanden:

X : device 8 write protect on
Y : device 9 write protect on
Z : device 8 write protect off

De 1526 printer is in feite een aangepaste CHINWA CTI CP 80 printer. Aan-
gezien een technische handleiding van de 1526 moeilijk verkrijgbaar is,
kan die van de SHINWA worden gebruikt. (m.u.v. voor wat betreft de spe-
cifieke aanpassingen voor de CBM-64)

Voor de 1541 drives zijn reeds meerdere versies van de E/F DISK ROMS ver-
schenen onder de nummers:

901229-02

901229-03

901229-05

Nummer 5 is de nieuwste. Hieruit zouden alle BUGS (fouten) gehaalt zijn.

De TV welke op Uw CBM-64 is aangesloten heeft de mogelijkheid om de
beeldkwaliteit te verbeteren. Door een aansluiting van een 2,2K potmeter
tussen pen 1 (luminance) en pen 4 (video out) kan men door verdraaien van
de potmeter de beeldkwaliteit op een normale TV verbeteren.

UITNODIGING BIJEENKOMST
=====

Datum : zaterdag 18 mei 1985
Lokatie : Soeeltuingebouw Beeklust/Ossenkoopeleerhoek
Jan Steenstraat 69. ALMELO. Tel.: 05490 - 19443

Reisroute :

- per auto - Vanuit het westen en het zuiden via de A1/A35.
1. Aan het einde van de snelweg rechtsaf. Bij de eerstvolgende stoplichten rechtdoor. Na ca. 100 m. kruispunt zonder stoplichten. Ga hier linksaf.
 2. Deze straat, de Jan Vermeerstraat, maakt een flauwe bocht naar rechts. Aan het einde van deze bocht ziet U links de lichtmasten van een voetbalveld. Ga linksaf. Het soeeltuingebouw vindt U na ca. 150 m. aan de linkerkant.
 3. Vanuit het noorden via de N36.
Rij door tot ANWB-borden richting Wierden/Zwolle. Ga hier rechtsaf. Blijf deze weg volgen. U komt dan over een spoorwegovergang.
Ga na ca. 150 m. linksaf, weer richting Wierden/Zwolle. Na ca. 200 m. ziet U rechts een Texaco-benzinestation. Ga hier rechtsaf. Verder als beschreven bij punt 2.

TOEGANGSPRIJS : FL. 10.==. Lunchoakket eventueel zelf mee te nemen.
Lunch is ook tegen vergoeding verkrijgbaar.
Konsumties tegen betaling.

PROGRAMMA : 09.30 Zaal open.
10.15 Doenino door de gastheren Erwin Visschedijk en Adri Hankel en door diens echtgenoten, die voor een uiterst vriendelijke bediening en begeleiding zorgen.
10.30 SAMSON 65 DOS COMPUTER en de toekomst.

De heer Loys Nachtman van het welbekende tijdschrift Elektuur over de inmiddels in Duitsland succesvolle SAMSON 65 Dos-computer met OHIO Dos OS65D V3.3.
Er zal niet alleen de computer gedemonstreerd worden, de heer Nachtman zal tevens iets vertellen over de toekomstige ontwikkelingen dienaangaande.

11.30 Koffiepauze.

11.45 Forum. Aan het forum kunnen vragen gesteld worden van allerlei aard.

12.00 Lunchoauze.

13.00 INFORMEEL GEDEELTE.

Tijdens het informeel gedeelte kunnen leden vrij met elkaar en met elkaars ervaringen kennis maken. Leden brengen hun systemen mee en demonstreren dit aan de aanwezigen. NEEM DAAROM UW COMPUTER MEE !!!

M A R K T. Op eigen tafel(s) te regelen.

17.00 Sluiting.

PMDL3 MICRO-WARE ASSEMBLER 65XX-1.1 221184 PAGE 01

```

0010:          GREEDY DEEL 3 (SLOT)
0020:
0030:          PHONS BLOEMEN
0040:          VINKELAAN 200
0050:          5702 LX HELMOND
0060:
0070: 8800          GREEDY ORG      $8800
0080:
0090:          *** KLADBLOK-RAM ***
0100:
0110:          00 00  DISPL  *      $0000
0120:          02 00  VELDPL *      $0002  VECTOR IN VELDGEGEVENS RAM
0130:          04 00  PRPNTL *      $0004  PRINT-VECTOR
0140:          08 00  JMIL   *      $0008  GEBRUIKT VOOR JMI
0150:          12 00  MAFLAG *      $0012  VOORBIJ IS
0160:          1A 00  MATIM  *      $001A
0170:          20 00  HCOORDL *     $0020  HULPCOORDINATEN I
0180:          22 00  MCOORDL *     $0022  HULPCOORDINATEN II
0190:          24 00  PMPOS  *      $0024  POSITIE VAN GREEDY
0200:          26 00  MAPOS  *      $0026  POSITIE MONSTER A
0210:          40 00  DIST   *      $0040  AFSTANDCODE
0220:          41 00  CHONE  *      $0041
0230:          42 00  CHTWO  *      $0042
0240:          47 00  EGGNUM *      $0047
0250:          48 00  EGGFLG *      $0048
0260:          52 00  DIRECA *      $0052  RICHTINGSBYTE VOOR DIREC.
0270:          53 00  CTRLA  *      $0053  CONTROLBYTES
0280:          55 00  CTRLB  *      $0055
0290:          5A 00  CHAR   *      $005A
0300:          5B 00  MONBEH *      $005B
0310:          5C 00  DIRSTA *      $005C
0320:          5D 00  DIRECE *      $005D
0330:          60 00  MASTAT *      $0060  STATUS MONSTER A J
0340:          65 00  MASPED *      $0065  SNELHEID MONSTER A J
0350:          6A 00  MADIR  *      $006A  LOOPRICHTING MONSTER A
0360:          6F 00  MAVER  *      $006F  STARTVERTRAGING MONSTER A
0370:          74 00  MACHAR *      $0074
0380:
0390:          *** PLAATSEN IN VIDEO RAM ***
0400:
0410:          F8 D1  HSCORD *      $D1F8  HI-SCORE
0420:          F8 D2  SCORD  *      $D2F8  SCORE
0430:          9E D2  MASTR  *      $D29E  STARTPLAATS MONSTER A
0440:          54 D3  PMSTR  *      $D354
0450:          5E D3  MESTR  *      $D35E
0460:          06 D1  POWA   *      $D106
0470:          2E D1  POWB   *      $D12E
0480:          86 D5  POWC   *      $D586
0490:          AE D5  POWD   *      $D5AE
0500:
0510:          86 85  SCRTAB *      $8586
0520:          86 87  TABEND *      $8786
0530:
0540:          32 88  ENDGAM *      $8832
0550:          70 88  NXTSTA *      $8870
0560:          65 89  PRINT  *      $8965
0570:          BA 89  CLSCR  *      $89BA
0580:          3C 8A  MAFIN  *      $8A3C
0590:          43 8A  MBFIN  *      $8A43
0600:          4A 8A  MCFIN  *      $8A4A
0610:          51 8A  MDFIN  *      $8A51
0620:          58 8A  MEFIN  *      $8A58
0630:          9C 8B  INPOIN *      $8B9C
0640:          B1 8B  DISVEC *      $8BB1

```

GREEDY MICRO-WARE ASSEMBLER 65XX-1.1 221184 PAGE 02

```

0650: BA 8B VELVEC * $8BBA
0660: C3 8B RECCHA * $8BC3
0670: 9C 8C EMPTY * $8C9C
0680: 9A 8D MDOWN * $8D9A
0690: F2 8D DEAD * $8DF2
0700: 83 8E MOVE * $8E83
0710: BA 8E TUNCHK * $8EBA
0720: 2E 8F DEPOIN * $8F2E
0730: C4 8F COMPAR * $8FC4
0740: D8 8F TURN * $8FD8
0750:
0760: 9000 PMDL3 ORG $9000
0770:
0780: *** HISCORE ROUTINE ***
0790:
0800: 9000 A9 86 NMWRT LDAIM TABEND
0810: 9002 85 02 STA VELDPL
0820: 9004 A9 87 LDAIM TABEND /
0830: 9006 85 03 STA VELDPL +01
0840: 9008 A9 FF LDAIM $FF BEGIN ONDERAAN CONTROLE =FF
0850: 900A 85 53 STA CTRLA
0860: 900C E6 53 NMA INC CTRLA NEEM SCORE VAN EEN PLAATS HOGER
0870: 900E 38 SEC
0880: 900F A5 02 LDA VELDPL PAS POINTER AAN
0890: 9011 85 08 STA JMIL EN BEWAAR HEM
0900: 9013 E9 40 SBCIM $40
0910: 9015 85 02 STA VELDPL
0920: 9017 A5 03 LDA VELDPL +01
0930: 9019 85 0C STA JMIL +01
0940: 901B E9 00 SBCIM $00
0950: 901D 85 03 STA VELDPL +01
0960: 901F 20 C4 8F JSR COMPAR VERGELIJK SCORES
0970: 9022 90 11 BCC NMC LAGER ?
0980: 9024 A0 00 LDYIM $00 NEE. SCHUIF SCORE IN TABEL
0990: 9026 B1 02 NAMB LDAIY VELDPL EEN PLAATS NAAR BENEDEN
1000: 9028 91 08 STAIY JMIL
1010: 902A C8 INY
1020: 902B C0 19 CPYIM $19 ALLES GEHAD ?
1030: 902D D0 F7 BNE NAMB
1040: 902F A5 53 LDA CTRLA ZIJN WE AL BIJ DE EERSTE PLAATS ?
1050: 9031 C9 08 CMPIM $08
1060: 9033 D0 D7 BNE NMA NEE. DOORGAAN
1070: 9035 A5 53 NMC LDA CTRLA ZIJN WE OP PLAATS 8 BLIJVEN STEKEN
1080: 9037 C9 01 CMPIM $01
1090: 9039 D0 03 BNE NMD
1100: 903B 4C 32 88 JMP ENDGAM SPEL AFGELOPEN
1110:
1120: *** NAAM SCHRIJVEN ***
1130:
1140: 903E A0 00 NMD LDYIM $00 BRENG SCORE OVER
1150: 9040 B9 F8 D2 NME LDAAY SCORD VAN SCHERM IN GEHEUGEN
1160: 9043 91 08 STAIY JMIL
1170: 9045 C8 INY
1180: 9046 C0 06 CPYIM $06
1190: 9048 D0 F6 BNE NME
1200: 904A 20 BA 89 JSR CLSCR SCHERM SCHOON
1210: 904D A9 11 LDAIM $11 PRESET 'CURSOR'
1220: 904F 85 00 STA DISPL
1230: 9051 A9 D1 LDAIM $D1
1240: 9053 85 01 STA DISPL +01
1250: 9055 20 65 89 JSR PRINT
1260: 9058 2A = '
1270: 9059 2A = '
1280: 905A 2A = '

```

| | | | | |
|-------|------|-------|----------------|----------------------------------|
| 1290: | 905B | 2A | = | '* |
| 1300: | 905C | 20 | = | ' |
| 1310: | 905D | 4E | = | 'N |
| 1320: | 905E | 41 | = | 'A |
| 1330: | 905F | 4D | = | 'M |
| 1340: | 9060 | 45 | = | 'E |
| 1350: | 9061 | 20 | = | ' |
| 1360: | 9062 | 52 | = | 'R |
| 1370: | 9063 | 45 | = | 'E |
| 1380: | 9064 | 47 | = | 'G |
| 1390: | 9065 | 49 | = | 'I |
| 1400: | 9066 | 53 | = | 'S |
| 1410: | 9067 | 54 | = | 'T |
| 1420: | 9068 | 52 | = | 'R |
| 1430: | 9069 | 41 | = | 'A |
| 1440: | 906A | 54 | = | 'T |
| 1450: | 906B | 49 | = | 'I |
| 1460: | 906C | 4F | = | 'O |
| 1470: | 906D | 4E | = | 'N |
| 1480: | 906E | 20 | = | ' |
| 1490: | 906F | 2A | = | '* |
| 1500: | 9070 | 2A | = | '* |
| 1510: | 9071 | 2A | = | '* |
| 1520: | 9072 | 2A | = | '* |
| 1530: | 9073 | 09 | = | \$09 |
| 1540: | 9074 | E5 | = | \$E5 SLA VIER REGELS OVER |
| 1550: | 9075 | 59 | = | 'Y |
| 1560: | 9076 | 4F | = | 'O |
| 1570: | 9077 | 55 | = | 'U |
| 1580: | 9078 | 20 | = | ' |
| 1590: | 9079 | 4D | = | 'M |
| 1600: | 907A | 55 | = | 'U |
| 1610: | 907B | 53 | = | 'S |
| 1620: | 907C | 54 | = | 'T |
| 1630: | 907D | 20 | = | ' |
| 1640: | 907E | 54 | = | 'T |
| 1650: | 907F | 59 | = | 'Y |
| 1660: | 9080 | 50 | = | 'P |
| 1670: | 9081 | 45 | = | 'E |
| 1680: | 9082 | 20 | = | ' |
| 1690: | 9083 | 31 | = | '1 |
| 1700: | 9084 | 36 | = | '6 |
| 1710: | 9085 | 20 | = | ' |
| 1720: | 9086 | 43 | = | 'C |
| 1730: | 9087 | 48 | = | 'H |
| 1740: | 9088 | 41 | = | 'A |
| 1750: | 9089 | 52 | = | 'R |
| 1760: | 908A | 41 | = | 'A |
| 1770: | 908B | 43 | = | 'C |
| 1780: | 908C | 54 | = | 'T |
| 1790: | 908D | 45 | = | 'E |
| 1800: | 908E | 52 | = | 'R |
| 1810: | 908F | 53 | = | 'S |
| 1820: | 9090 | 03 | = | \$03 |
| 1830: | 9091 | A9 12 | LDAIM \$12 | PRESET 'CURSOR' |
| 1840: | 9093 | 85 04 | STA PRPNTL | |
| 1850: | 9095 | A9 D4 | LDAIM \$D4 | |
| 1860: | 9097 | 85 05 | STA PRPNTL +01 | |
| 1870: | 9099 | A0 00 | LDYIM \$00 | BRENG SCORE WEER NAAR HET SCHERM |
| 1880: | 909B | B1 0B | LDAIY JMIL | |
| 1890: | 909D | 91 04 | STAIY PRPNTL | |
| 1900: | 909F | C8 | INY | |
| 1910: | 90A0 | C0 06 | CPYIM \$06 | OVERGEBRACHT ? |
| 1920: | 90A2 | D0 F7 | BNE NMF | |

NMF

```

1930: 90A4 C8          INY          2 SPATIES
1940: 90A5 C8          INY
1950: 90A6 20 C3 8B NMG JSR RECCHA HAAL KARAKTER OP
1960: 90A9 C9 7F      CMPIM $7F  DELETE ?
1970: 90AB F0 0C      BEQ NMH
1980: 90AD 91 0B      STAIY JML  ZET IN GEHEUGEN
1990: 90AF 91 04      STAIY PRPNTL EN OP SCHERM
2000: 90B1 C8          INY
2010: 90B2 C0 18      CPYIM $18  VOL ?
2020: 90B4 D0 F0      BNE NMG
2030: 90B6 4C 32 8B JMP ENDGAM
2040: 90B9 A9 20      NMH LDAIM $20  WIS VORIG KARAKTER
2050: 90BB 88          DEY
2060: 90BC 91 0B      STAIY JML  IN GEHEUGEN
2070: 90BE 91 04      STAIY PRPNTL EN OP SCHERM
2080: 90C0 4C A6 90 JMP NMG EN GA DOOR
2090:
2100: *** LOPENDE MONSTERS ***
2110:
2120: 90C3 A2 00      MAROUT LDXIM $00  MONSTER A
2130: 90C5 20 EB 90 JSR MRT HARTEN
2140: 90C8 4C 3C 8A JMP MAFIN
2150: 90CB A2 01      MBROUT LDXIM $01  MONSTER B
2160: 90CD 20 EB 90 JSR MRT KLAVEREN
2170: 90D0 4C 43 8A JMP MBFIN
2180: 90D3 A2 02      MCROUT LDXIM $02  MONSTER C
2190: 90D5 20 EB 90 JSR MRT SCHOPPEN
2200: 90D8 4C 4A 8A JMP MCFIN
2210: 90DB A2 03      MDROUT LDXIM $03  MONSTER D
2220: 90DD 20 EB 90 JSR MRT RUITEN
2230: 90E0 4C 51 8A JMP MDFIN
2240: 90E3 A2 04      MEROUT LDXIM $04  MONSTER E
2250: 90E5 20 EB 90 JSR MRT INVERSE RUITEN
2260: 90E8 4C 58 8A JMP MEFIN
2270:
2280: *** MONSTER RUOTINE VOOR ALLE 5 ***
2290:
2300: 90EB 86 5B      MRT STX MONBEH BEWAAR MONSTER DAT BEHANDELD WORDT
2310: 90ED B5 60      LDAAX MASTAT MOET HET WEL WORDEN BEHANDELD ?
2320: 90EF 29 7F      ANDIM $7F
2330: 90F1 D0 01      BNE MR STATUS 0 DOET NIET MEE
2340: 90F3 60          RTS
2350: 90F4 B5 6F      MR LDAAX MAVER IS STARTVERTRAGING AL VORBIJ ?
2360: 90F6 F0 34      BEQ WALK DAN LOPE
2370: 90F8 D6 6F      DECAX MAVER EEN MINDER
2380: 90FA D0 1A      BNE SIT VOORBIJ ?
2390: 90FC 20 F8 92 JSR XYUP ZET X EN Y OP
2400: 90FF A9 5E      LDAIM MESTR ZET MONSTER OP STARTPLAATS
2410: 9101 85 20      STA HCOORL
2420: 9103 A9 D3      LDAIM MESTR /
2430: 9105 85 21      STA HCOORL +01
2440: 9107 A9 80      LDAIM $80 GEEF HET RICHTING $80
2450: 9109 99 6A 00 STAIY MADIR (ONMOGELIJK BIJ START)
2460: 910C A9 20      LDAIM $20 LAAT SPATIE ACHTER
2470: 910E 99 74 00 STAIY MACHAR
2480: 9111 85 5A      STA CHAR
2490: 9113 20 D3 92 JSR MONDIS ZET OP SCHERM
2500: 9116 A6 5B      SIT LDX MONBEH ZET VERTRAGINGEN WEER OP
2510: 9118 B5 60      LDAAX MASTAT KWETSBARE MONSTERS LOPE LANGZAMER
2520: 911A C9 80      CMPIM $80 BIT 7 GESET ?
2530: 911C 10 05      BPL SI
2540: 911E B5 65      LDAAX MASPED SNELHEID MONSTER
2550: 9120 4C 25 91 JMP ST
2560: 9123 A9 40      SI LDAIM $40

```

PMDL3 MICRO-WARE ASSEMBLER 65XX-1.1 221184 PAGE 05

```

2570: 9125 95 1A      ST      STAAX MATIM  ZET VERTRAGING
2580: 9127 A9 00      LDAIM $00    RESET VLAG
2590: 9129 95 12      STAAX MAFLAG
2600: 912B 60      RTS
2610:
2620:                *** LOOP ROUTINE ***
2630:
2640: 912C 20 F8 92    WALK    JSR    XYUP    ZET X EN Y OP
2650: 912F B5 26      LDAAX MAPOS  KOPIEER KOORDINATEN IN HCOORL
2660: 9131 85 20      STA      HCOORL
2670: 9133 B5 27      LDAAX MAPOS  +01
2680: 9135 85 21      STA      HCOORL +01 MAAK ER VECTOR IN VELD-
2690: 9137 20 BA 8B    JSR    VELVEC GEGEVENS RAM VAN
2700: 913A 20 B6 92    JSR    CROSS  STAAT MONSTER OP KRUISING ?
2710: 913D 90 03      BCC     NKRUS  GEEN KRUISING
2720: 913F 4C 54 92    JMP     CHDIR  KIES RICHTING
2730: 9142 A6 5B      NKRUS   LDX     MONBEH
2740: 9144 A0 00      LDYIM $00    KIK OF MONSTER NOG STEEDS
2750: 9146 B1 20      LDAIY HCOORL IN ZIJN RICHTING KAN
2760: 9148 29 F0      ANDIM $F0    DOORLOPEN / ALLEEN BIT 4-5-6-7
2770: 914A 35 6A      ANDAX MADIR
2780: 914C D0 03      BNE     GOON
2790: 914E 4C 54 92    JMP     CHDIR  KIES ANDERE RICHTING
2800:
2810:                *** VERPLAATS MONSTER ***
2820:
2830: 9151 A5 52      GOON     LDA     DIRECA BEWAAR DIRECA (PM RICHTING)
2840: 9153 85 5D      STA     DIRECE
2850: 9155 B5 6A      LDAAX  MADIR  HAAL MONSTER RICHTING
2860: 9157 85 52      STA     DIRECA PREPAREER VOOR MOVE
2870: 9159 20 B1 8B    JSR    DISVEC MAAK WEER DISPLAY VECTOR
2880: 915C 20 83 8E    JSR    MOVE   VERPLAATS
2890: 915F 20 BA 8E    JSR    TUNCHK LET OP TUNNELS
2900: 9162 A5 5D      LDA     DIRECE ZET DIRECA WEER GOED
2910: 9164 85 52      STA     DIRECA
2920: 9166 A5 24      LDA     PMPOS  OP PLAAT VAN GREEDY TERECHTGEKOMEN ?
2930: 9168 C5 20      CMP     HCOORL DAN OPETEN OF OPGETEN WORDEN
2940: 916A D0 06      BNE     WHAT
2950: 916C A5 25      LDA     PMPOS  +01
2960: 916E C5 21      CMP     HCOORL +01
2970: 9170 F0 54      BEQ     ONTMOE
2980: 9172 A0 00      WHAT    LDYIM $00    KIK WAAR MONSTER TERECHTKOMT
2990: 9174 A2 00      LDXIM $00
3000: 9176 B1 20      LDAIY HCOORL
3010: 9178 C9 07      CMPIM $07  HELEMAAL NIETS ?
3020: 917A 90 66      BCC     PLUSON DAN EEN OVERGANG BIJTellen
3030: 917C DD 96 91    DETER   CMPAX  TABLA  GA TABEL AF
3040: 917F F0 08      BEQ     INLI
3050: 9181 E8      INX
3060: 9182 E0 10      CPXIM $10
3070: 9184 D0 F6      BNE     DETER
3080: 9186 4C E0 91    JMP     ILLEG  DAN VLEKJE. WEGWERKEN
3090: 9189 BC A6 91    INLI    LDYAX  TABLB  HAAL ADRES BIJBEHORENDE ROUTINE OP
3100: 918C 84 0B      STY     JMIL
3110: 918E BC B6 91    LDYAX  TABLC
3120: 9191 84 0C      STY     JMIL +01
3130: 9193 6C 0B 00    JMI     JMIL
3140: 9196 E5      TABLA  =     $E5  MONSTER ONTMOETINGEN
3150: 9197 E6      =     $E6
3160: 9198 E7      =     $E7
3170: 9199 E8      =     $E8
3180: 919A E9      =     $E9
3190: 919B BB      =     $BB  OOK KWETSBAAR
3200: 919C B7      =     $B7

```

PMDL3 MICRO-WARE ASSEMBLER 65XX-1.1 221184 PAGE 06

| | | | | | |
|-------|------|----------|--------------|----------|-------------------------------------|
| 3210: | 919D | B8 | = | \$B8 | BARRICADES |
| 3220: | 919E | 2E | = | \$2E | KLEINE PUNT |
| 3230: | 919F | 07 | = | \$07 | NIEUWE PUNT |
| 3240: | 91A0 | AB | = | \$AB | DIKKE PUNT |
| 3250: | 91A1 | E2 | = | \$E2 | POWER |
| 3260: | 91A2 | 2A | = | \$2A | STER |
| 3270: | 91A3 | EA | = | \$EA | SLETTEL |
| 3280: | 91A4 | EB | = | \$EB | |
| 3290: | 91A5 | 10 | = | \$10 | DUMMY |
| 3300: | 91A6 | 28 | TABLB = | BARICA | |
| 3310: | 91A7 | 28 | = | BARICA | |
| 3320: | 91A8 | 28 | = | BARICA | |
| 3330: | 91A9 | 28 | = | BARICA | |
| 3340: | 91AA | 28 | = | BARICA | |
| 3350: | 91AB | 28 | = | BARICA | |
| 3360: | 91AC | 28 | = | BARICA | |
| 3370: | 91AD | 28 | = | BARICA | |
| 3380: | 91AE | E0 | = | ILLEG | LATEN VERDWIJNE |
| 3390: | 91AF | 00 | = | NPUNT | |
| 3400: | 91B0 | 3F | = | DPUNT | |
| 3410: | 91B1 | E5 | = | STORE | OPBERGEN EN VERDER NIETS |
| 3420: | 91B2 | E5 | = | STORE | |
| 3430: | 91B3 | E5 | = | STORE | |
| 3440: | 91B4 | E5 | = | STORE | |
| 3450: | 91B5 | E5 | = | STORE | |
| 3460: | 91B6 | 92 | TABLC = | BARICA / | |
| 3470: | 91B7 | 92 | = | BARICA / | |
| 3480: | 91B8 | 92 | = | BARICA / | |
| 3490: | 91B9 | 92 | = | BARICA / | |
| 3500: | 91BA | 92 | = | BARICA / | |
| 3510: | 91BB | 92 | = | BARICA / | |
| 3520: | 91BC | 92 | = | BARICA / | |
| 3530: | 91BD | 92 | = | BARICA / | |
| 3540: | 91BE | 91 | = | ILLEG / | |
| 3550: | 91BF | 92 | = | NPUNT / | |
| 3560: | 91C0 | 92 | = | DPUNT / | |
| 3570: | 91C1 | 91 | = | STORE / | |
| 3580: | 91C2 | 91 | = | STORE / | |
| 3590: | 91C3 | 91 | = | STORE / | |
| 3600: | 91C4 | 91 | = | STORE / | |
| 3610: | 91C5 | 91 | = | STORE / | |
| 3620: | | | | | |
| 3630: | 91C6 | 20 F8 92 | ONTMOE JSR | XYUP | ZET X EN Y OP |
| 3640: | 91C9 | B9 60 00 | LDAAY | MASTAT | EETBAAR-VLAG AAN ? |
| 3650: | 91CC | C9 80 | CMPIM | \$80 | BIT 7 GESET ? |
| 3660: | 91CE | 10 05 | BPL | OPETEN | |
| 3670: | 91D0 | 68 | PLA | | NOODUITGANG UIT SUBROUTINE |
| 3680: | 91D1 | 68 | JMP | | |
| 3690: | 91D2 | 4C F2 8D | JMP | DEAD | MONSTER EET GREEDY OP |
| 3700: | 91D5 | B9 74 00 | OPETEN LDAAY | MACHAR | ZET KARAKTER WAAR MONSTER |
| 3710: | 91D8 | 81 26 | STAIX | MAPOSL | OP STOND TERUG |
| 3720: | 91DA | 20 9A 8D | JSR | MDOWN | VOER OPEETROUTINE UIT |
| 3730: | 91DD | 4C 16 91 | JMP | SIT | |
| 3740: | | | | | |
| 3750: | 91E0 | A9 00 | ILLEG LDAIM | \$00 | VERWIJDE VLEKJE |
| 3760: | 91E2 | 18 | PLUSON CLC | | TEL EEN MONSTERVERGANG BIJ |
| 3770: | 91E3 | 69 01 | ADCIM | \$01 | |
| 3780: | 91E5 | 48 | STORE PHA | | BEWAAR |
| 3790: | 91E6 | A6 5B | LDX | MONBEH | |
| 3800: | 91E8 | B5 74 | LDAAX | MACHAR | HAAL KARAKTER WAAR MONSTER EERST OP |
| 3810: | 91EA | 85 5A | STA | CHAR | STOND OM HET TERUG TE ZETTEN |
| 3820: | 91EC | C9 AB | CMPIM | \$AB | NIEUWE DIKKE PUNT ? |
| 3830: | 91EE | D0 03 | BNE | NDIK | DAN PUNTENTELLER NU PAS VERHOGEN ! |
| 3840: | 91F0 | 20 9C 8B | JSR | INPOIN | |

PMDL3 MICRO-WARE ASSEMBLER 65XX-1.1 221184 PAGE 07

```

3850: 91F3 20 F8 92 NDIK JSR XYUP ZET X EN Y OP
3860: 91F6 68 PLA KARAKTER WAAR MONSTER
3870: 91F7 99 74 00 STAAY MACHAR OP GAAT STAAN
3880: 91FA 20 D3 92 JSR MONDIS VERPLAATS MONSTER OP SCHERM
3890: 91FD 4C 16 91 JMP SIT EN ZET VERTRAGINGEN WEER OP
3900:
3910: 9200 A5 21 NPUNT LDA HCOORL +01 GEEN PUNT OP STARTPLAATSEN
3920: 9202 C9 D3 CMPIM PMSTR / VAN GREEDY EN MONSTERS
3930: 9204 D0 0A BNE NIEUW
3940: 9206 A5 20 LDA HCOORL
3950: 9208 C9 54 CMPIM PMSTR
3960: 920A F0 D4 BEQ ILLEG ZET ER WEER $00
3970: 920C C9 5E CMPIM MESTR
3980: 920E F0 D0 BEQ ILLEG
3990: 9210 E6 47 NIEUW INC EGGNUM 1 PUNT GEPRODUCEERD
4000: 9212 A5 47 LDA EGGNUM MAX 176 EIEREN LEGGEN
4010: 9214 C9 B0 CMPIM $B0
4020: 9216 F0 09 BEQ EGGUP
4030: 9218 A5 48 LDA EGGFLG MAG EI GELEGD WORDEN ?
4040: 921A D0 C4 BNE ILLEG
4050: 921C A9 AB LDAIM $AB MAAK NIEUW DIK PUNT
4060: 921E 4C E5 91 JMP STORE
4070: 9221 A9 01 EGGUP LDAIM $01 ZET DE VLAG
4080: 9223 B5 48 STA EGGFLG
4090: 9225 4C E0 91 JMP ILLEG
4100:
4110: 9228 20 F8 92 BARICA JSR XYUP
4120: 922B B9 6A 00 LDAAY MADIR RICHTING IN MADIR
4130: 922E 49 F0 EORIM $F0 IS NU NIET MEER MOGELIJK
4140: 9230 25 5C AND DIRSTA DUS BITJE SCHRAPPEN UIT DIRSTA
4150: 9232 B5 5C STA DIRSTA
4160: 9234 B5 26 LDAAX MAPOS
4170: 9236 B5 20 STA HCOORL HERSTEL OUDE HCOORL
4180: 9238 B5 27 LDAAX MAPOS +01
4190: 923A B5 21 STA HCOORL +01
4200: 923C 4C 6F 92 JMP LUCKY KIES NIEUWE RICHTING
4210:
4220: 923F 20 2E 8F DPUNT JSR DEPOIN TREK EEN PUNT VAN PUNTENTELLER AF
4230: 9242 B0 05 BCS VLDEPT VELD LEEG ?
4240: 9244 A9 2E LDAIM $2E MAAK ER EEN DUN PUNT VAN
4250: 9246 4C E5 91 JMP STORE
4260: 9249 68 VLDEPT PLA NOODUITGANG UIT SUBROUTINE
4270: 924A 68 PLA
4280: 924B A9 00 LDAIM $00 VERWIJDER PUNT
4290: 924D A0 00 LDYIM $00
4300: 924F 91 20 STAIY HCOORL
4310: 9251 4C 9C 8C JMP EMPTY A=0, 7 X BONUS
4320:
4330: *** KEUZE NIEUWE RICHTING ***
4340:
4350: 9254 A6 5B CHDIR LDX MONBEH
4360: 9256 B5 6A LDAAX MADIR ELIMINEER RICHTING TEGENOVERGESTELD
4370: 9258 20 D8 8F JSR TURN AAN OUDE RICHTING VAN MONSTER
4380: 925B 49 F0 EORIM $F0
4390: 925D 25 5C AND DIRSTA
4400: 925F B5 5C STA DIRSTA
4410: 9261 B5 60 LDAAX MASTAT HAAL STATUS MONSTER OP
4420: 9263 C9 80 CMPIM $80 EETBAAR MONSTER ?
4430: 9265 10 08 BPL LUCKY ZOMAAR LATEN LOPEN
4440: 9267 C9 04 CMPIM $04 HOE HOGER DE STATUS
4450: 9269 10 30 BPL TRACE DES TE SLIMMER HET MONSTER
4460: 926B C9 02 CMPIM $02
4470: 926D 10 29 BPL GUARD
4480:

```

PMDL3 MICRO-WARE ASSEMBLER 65XX-1.1 221184 PAGE 08

```

4490:          *** RANDOM LATEN LOPEN ***
4500:
4510: 926F 20 A7 92 LUCKY JSR  RANDIR HAAL ZOMAAR EEN RICHTING
4520: 9272 A5 5C LDA  DIRSTA KIJK OF DEZE KAN
4530: 9274 F0 1F BEQ  IMPOS ZOU HELEMAAL NIET GEKUND HEBBEN
4540: 9276 A5 5D TRY  LDA  DIRECE GEKOZEN RICHTING
4550: 9278 25 5C AND  DIRSTA MOGELIJK ?
4560: 927A D0 05 BNE  KLAAR
4570: 927C 06 5D ASL  DIRECE BLIJF NET ZOLANG SCHUIVEN TOT HET KAN
4580: 927E 4C 76 92 JMP  TRY
4590: 9281 A6 5B KLAAR LDX  MONBEH ZET DEZE RICHTING IN MADIR
4600: 9283 A5 5D LDA  DIRECE
4610: 9285 95 6A STAAX MADIR
4620: 9287 20 BA 8B JSR  VELVEC VELDGEGEVENS VECTOR
4630: 928A A5 5D LDA  DIRECE KIJK OF DE RICHTING ECHT WEL KAN
4640: 928C A0 00 LDYIM $00
4650: 928E 31 20 ANDIY HCOORL
4660: 9290 F0 03 BEQ  IMPOS
4670: 9292 4C 51 91 JMP  GOON
4680: 9295 4C 16 91 IMPOS JMP  SIT DOE HELEMAAL NIETS
4690:
4700: 9298 4C FE 92 GUARD JMP  GA
4710: 929B 4C 57 93 TRACE JMP  TRA
4720:
4730:          *** SUBROUTINES ***
4740:
4750:          RANDOM : Kiest een getal tussen 0 en 3
4760:
4770: 929E AD 04 18 RANDOM LDA  $1804 LEES VIA TIMER REGISTER UIT
4780: 92A1 4D 08 18 EOR  $1808
4790: 92A4 29 03 ANDIM $03
4800: 92A6 60 RTS
4810:
4820:          RANDIR : Kiest een willekeurige richting
4830:
4840: 92A7 20 9E 92 RANDIR JSR  RANDOM HAAL GETAL
4850: 92AA AA TAX  NAAR X
4860: 92AB 38 SEC
4870: 92AC A9 08 LDAIM $08 CONFIGURATIE 00001000 1
4880: 92AE 85 5D STA  DIRECE ROTEREN TOT X LEEG IS
4890: 92B0 26 5D ROT  ROL DIRECE
4900: 92B2 CA DEX
4910: 92B3 10 FB BPL  ROT
4920: 92B5 60 RTS
4930:
4940:          CROSS : Kijk of monster op kruising zit
4950:
4960: 92B6 A0 00 CROSS LDYIM $00
4970: 92B8 B1 20 LDAIY HCOORL HAAL RICHTINGENPLAATJE
4980: 92BA 29 F0 ANDIM $F0 ALLEEN BIT 4-5-6-7
4990: 92BC 85 5C STA  DIRSTA MOGELIJKE RICHTINGEN
5000: 92BE 20 C9 92 JSR  BITNUM KIJK HOEVEEL ER MOGELIJK ZIJN
5010: 92C1 C0 03 CPYIM $03 MEER DAN 3 RICHTINGEN ?
5020: 92C3 10 02 BPL  CRC
5030: 92C5 18 CLC  C=0
5040: 92C6 60 RTS
5050: 92C7 38 CRC  SEC  C=1
5060: 92C8 60 RTS
5070:
5080:          BITNUM : Kijkt hoeveel bits in A gezet zijn
5090:
5100: 92C9 A2 08 BITNUM LDXIM $08
5110: 92CB 0A BITA ASLA SCHUIF
5120: 92CC 90 01 BCC  BITB

```

PMDL3 MICRO-WARE ASSEMBLER 65XX-1.1 221184 PAGE 09

```

5130: 92CE C8      INY      Y IS BITTELLER
5140: 92CF CA      BITB    DEX      8 X GESCHOVEN ?
5150: 92D0 10 F9    BPL      BITA
5160: 92D2 60      RTS
5170:
5180:
5190:
MONDIS : VERPLAATS MONSTER OP SCHERM
5200: 92D3 20 F8 92 MONDIS JSR      XYUP    ZET X EN Y OP
5210: 92D6 20 B1 8B JSR      DISVEC  HET MONSTER STAAT IN HCOORL
5220: 92D9 A5 5A    LDA      CHAR    KARAKTER WAAR MONSTER OP STOND
5230: 92DB 81 26    STAIX   MAOSL    WIS MONSTER ERMEE
5240: 92DD A5 20    LDA      HCOORL  BRENG NIEUWE COORDINATEN
5250: 92DF 95 26    STAAX   MAOSL    OVER IN MAPSL
5260: 92E1 A5 21    LDA      HCOORL  +01
5270: 92E3 95 27    STAAX   MAOSL    +01
5280: 92E5 B9 60 00 LDAAY   MASTAT  KIJK NAAR STATUS
5290: 92E8 C9 80    CMPIM   $80    GRIJS MAKEN ?
5300: 92EA 10 07    BPL      GREY
5310: 92EC 98      TYA      BEPAAL FIGUURTJE
5320: 92ED 18      CLC
5330: 92EE 69 E5    ADCIM   $E5    DOOR $E5 OP TE TELLEN
5340: 92F0 4C F5 92 JMP      FIG
5350: 92F3 A9 BB    GREY   LDAIM   $BB
5360: 92F5 81 26    FIG     STAIX   MAOSL  ZET MONSTER OP SCHERM
5370: 92F7 60      RTS
5380:
5390:
5400:
XYUP : ZET MONBEH IN Y EN HET DUBBELE IN X
5410: 92F8 A5 5B    XYUP   LDA      MONBEH
5420: 92FA A8      TAY
5430: 92FB 0A      ASLA
5440: 92FC AA      TAX
5450: 92FD 60      RTS
5460:
5470:
5480:
*** BEWAAK ROUTINE VOOR POWERPILLEN ***
5490: 92FE A9 04    GA      LDAIM   $04    ER ZIJN MAX 4 POWERPILLEN
5500: 9300 85 53    STA      CTRLA
5510: 9302 A5 5B    LDA      MONBEH  OM NIET ALTIJD DEZELFDE
5520: 9304 29 03    WHICH  ANDIM   $03    TE NEMEN ALS EERSTE
5530: 9306 AA      TAX
5540: 9307 BD 38 93 LDAAX   PWTAB    HAAL COORDINATEN VAN POWERPIL OP
5550: 930A 85 22    STA      MCOORL
5560: 930C BD 3C 93 LDAAX   PWTABB
5570: 930F 85 23    STA      MCOORL  +01
5580: 9311 A0 00    LDYIM   $00
5590: 9313 B1 22    LDAIY   MCOORL  STAAT DE PIL ER NOG ?
5600: 9315 C9 E2    CMPIM   $E2
5610: 9317 F0 27    BEQ      FND
5620: 9319 C6 53    DEC      CTRLA  ALLES GEPROBEERD ?
5630: 931B F0 05    BEQ      NOVER
5640: 931D E8      INX
5650: 931E 8A      TXA      PAK GEWOON DE VOLGENDE
5660: 931F 4C 04 93 JMP      WHICH
5670: 9322 A6 5B    NOVER  LDX      MONBEH  VERANDER STATUS
5680: 9324 B5 60    LDAAX   MASTAT
5690: 9326 C9 02    CMPIM   $02    STATUS 2 ?
5700: 9328 F0 07    BEQ      DOLOS
5710: 932A A9 05    LDAIM   $05    STATUS 3 WORDT STATUS 5
5720: 932C 95 60    STAAX   MASTAT
5730: 932E 4C 57 93 JMP      TRA
5740: 9331 A9 01    DOLOS  LDAIM   $01    STATUS 2 WORDT STATUS 1
5750: 9333 95 60    STAAX   MASTAT
5760: 9335 4C 6F 92 JMP      LUCKY

```

PMDL3 MICRO-WARE ASSEMBLER 65XX-1.1 221184 PAGE 10

```

5770: 9338 06      PWTAB = POWA TABEL MET COORDINATEN
5780: 9339 2E      = POWB VAN POWERPILLEN
5790: 933A 86      = POWC
5800: 933B AE      = POWD
5810: 933C D1      PWTABB = POWA /
5820: 933D D1      = POWB /
5830: 933E D5      = POWC /
5840: 933F D5      = POWD /
5850:
5860: 9340 20 96 93 FND JSR BEREK BEREKEN RICHTING NAAR PIL
5870: 9343 A5 41 CH LDA CHONE KIJK NAAR EERSTE KEUS
5880: 9345 25 5C AND DIRSTA MOGELIJK ?
5890: 9347 D0 09 BNE MOG
5900: 9349 A5 42 LDA CHTWO TWEDE KEUS
5910: 934B 25 5C AND DIRSTA
5920: 934D D0 03 BNE MOG
5930: 934F 4C 6F 92 JMP LUCKY DAN KIES ER ZOMAAR EEN
5940: 9352 85 5D MOG STA DIRECE NIEUWE RICHTING
5950: 9354 4C 81 92 JMP KLAAR LAATSTE CONTROLE
5960:
5970: *** ROUTINE OM GREEDY TE VOLGEN ***
5980:
5990: 9357 A5 24 TRA LDA PMPOSL ZET COORDINATEN GREEDY IN
6000: 9359 85 22 STA MCOORL MCOORL VOOR BEREKENEN
6010: 935B A5 25 LDA PMPOSL +01 RICHTING
6020: 935D 85 23 STA MCOORL +01
6030: 935F 20 96 93 JSR BEREK
6040: 9362 A6 5B LDX MONBEH
6050: 9364 B5 60 LDAAX MASTAT IS HET STATUS 4.
6060: 9366 C9 04 CMPIM $04 DE 'PLAKKER' ?
6070: 9368 F0 09 BEQ NDICHT
6080: 936A A5 40 LDA DIST MONSTER NIET TE DICHTBIJ.
6090: 936C C9 0F CMPIM $0F OM HET EERLIJK TE HOUDEN
6100: 936E B0 03 BCS NDICHT
6110: 9370 4C 6F 92 NTRACE JMP LUCKY DOE HET OP DE GOK
6120: 9373 B5 60 NDICHT LDAAX MASTAT KIJK NAAR DE STATUS
6130: 9375 A4 40 LDY DIST AFSTAND IN Y
6140: 9377 29 03 ANDIM $03 STATI 4-5-6-7 WORDEN 0-1-2-3
6150: 9379 F0 0A BEQ PLAK NR 4: PLAKKER
6160: 937B AA TAX
6170: 937C CA DEX
6180: 937D F0 0B BEQ TRACA NR 5 VOLGER TUSSEN 15 EN 35
6190: 937F CA DEX
6200: 9380 F0 0D BEQ TRACB NR 6 VOLGER TUSSEN 15 EN 70
6210: 9382 4C 93 93 JMP TRACC NR 7 VOLGER TUSSEN 15 EN ONEINDIG
6220: 9385 C0 0F PLAK CPYIM $0F NIET VERDER DAN 15
6230: 9387 4C 91 93 JMP RUIK
6240: 938A C0 23 TRACA CPYIM $23
6250: 938C 4C 91 93 JMP RUIK
6260: 938F C0 46 TRACB CPYIM $46
6270: 9391 10 DD RUIK BPL NTRACE BUITEN BEREIK ?
6280: 9393 4C 43 93 TRACC JMP CH KIES DE BESTE RICHTING
6290:
6300: *** BEREKEN SUBROUTINE VOOR GOEDE RICHTING ***
6310:
6320: 9396 A5 20 BEREK LDA HCOORL BRENG OVER OM TE BEWERKEN
6330: 9398 85 53 STA CTRLA
6340: 939A A5 21 LDA HCOORL +01
6350: 939C 29 07 ANDIM $07
6360: 939E 85 54 STA CTRLA +01 CTRLA : VERTICALE COORDINAAT
6370: 93A0 A5 22 LDA MCOORL CTRLA +01 : HORIZONTALE COORDINAAT
6380: 93A2 85 55 STA CTRLB GELDT OOK VOOR CTRLB
6390: 93A4 A5 23 LDA MCOORL +01
6400: 93A6 29 07 ANDIM $07

```

PMDL3 MICRO-WARE ASSEMBLER 65XX-1.1 221184 PAGE 11

```

6410: 93A8 85 56          STA CTRLB +01
6420: 93AA A2 06          LDXIM $06 6 X SCHUIVEN
6430: 93AC 46 54          VERCOR LSR CTRLA +01 TOTDAT ER EEN
6440: 93AE 66 53          ROR CTRLA VERTICALE COORDINAAT ONSTAAT
6450: 93B0 46 56          LSR CTRLB +01
6460: 93B2 66 55          ROR CTRLB
6470: 93B4 CA          DEX
6480: 93B5 10 F5          BPL VERCOR
6490: 93B7 A5 20          HORCOR LDA HCOORL BEPAAL HORIZONTALE COORDINAAT
6500: 93B9 29 3F          ANDIM $3F
6510: 93BB 85 54          STA CTRLA +01
6520: 93BD A5 22          LDA MCOORL
6530: 93BF 29 3F          ANDIM $3F
6540: 93C1 85 56          STA CTRLB +01
6550: 93C3 A5 55          BERA LDA CTRLB
6560: 93C5 38          SEC BEPAAL HET VERSCHIL TUSSEN
6570: 93C6 E5 53          SBC CTRLA DE COORDINATEN
6580: 93C8 90 10          BCC BERB NEGATIEF: NIETS DOEN
6590: 93CA 85 40          STA DIST POSITIEF: VERTICALE VERSCHILLEN
6600: 93CC 18          CLC DRIE MAAL TELLEN VOOR DO.
6610: 93CD 65 40          ADC DIST HORIZONTALE TWEE MAAL:
6620: 93CF 65 40          ADC DIST VERTICALE RICHTING IN CHONE ZETTEN.
6630: 93D1 85 40          STA DIST HORIZONTALE IN CHTWO
6640: 93D3 A9 20          LDAIM $20 OMLAAG
6650: 93D5 85 41          STA CHONE
6660: 93D7 4C EA 93      JMP BERB
6670: 93DA 38          SEC
6680: 93DB A5 53          LDA CTRLA
6690: 93DD E5 55          SBC CTRLB
6700: 93DF 85 40          STA DIST
6710: 93E1 18          CLC
6720: 93E2 65 40          ADC DIST
6730: 93E4 65 40          ADC DIST
6740: 93E6 A9 80          LDAIM $80 OMHOOG
6750: 93E8 85 41          STA CHONE
6760: 93EA 38          SEC
6770: 93EB A5 56          LDA CTRLB +01
6780: 93ED E5 54          SBC CTRLA +01
6790: 93EF 90 14          BCC BERD NEGATIEF ?
6800: 93F1 0A          ASLA VERDUBBEL
6810: 93F2 48          PHA BEWAAR OM TE KIJKEN WELKE RICHTING
6820: 93F3 A0 10          LDYIM $10 1E KEUS EN WELKE 2E KEUS
6830: 93F5 84 42          STY CHTWO MOET WORDEN
6840: 93F7 C5 40          CMP DIST NAAR RECHTS
6850: 93F9 90 03          BCC BERE
6860: 93FB 20 1E 94      JSR DROM OMWISSELEN
6870: 93FE 68          PLA BEPAAL DIST
6880: 93FF 18          CLC
6890: 9400 65 40          ADC DIST
6900: 9402 85 40          STA DIST
6910: 9404 60          RTS DAT WAS HET AN
6920: 9405 38          SEC
6930: 9406 A5 54          LDA CTRLA +01
6940: 9408 E5 56          SBC CTRLB +01
6950: 940A 0A          ASLA VERDUBBEL
6960: 940B 48          PHA BEWAAR EVEN, ZIE BOVEN
6970: 940C A0 40          LDYIM $40
6980: 940E 84 42          STY CHTWO
6990: 9410 C5 40          CMP DIST OMWISSELEN ?
7000: 9412 90 03          BCC BERF
7010: 9414 20 1E 94      JSR DROM
7020: 9417 68          PLA BEPAAL DIST
7030: 9418 18          CLC
7040: 9419 65 40          ADC DIST
7050: 941B 85 40          STA DIST
7060: 941D 60          RTS DAT WAS HET DAN
7070:
7080: DROM : VERWISSELT CHONE EN CHTWO
7090:
7100: 941E A6 41          DROM LDX CHONE
7110: 9420 A5 42          LDA CHTWO
7120: 9422 85 41          STA CHONE
7130: 9424 86 42          STX CHTWO
7140: 9426 60          RTS
7150:
7160: EINDE GREEDY
7170: deel 1 en 2 in voorgaande edities

```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 9600: | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | |
| 9610: | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | |
| 9620: | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | |
| 9630: | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | FF | EF | E0 | D0 | |
| 9640: | C0 | A4 | 98 | 8E | 86 | 7E | 75 | 6E | 62 | 5A | 53 | 4E | 49 | 44 | 40 | 3C | |
| 9650: | 38 | 34 | 30 | 2E | 2B | 28 | 25 | 23 | 20 | 20 | 20 | 31 | 30 | 20 | 31 | 32 | |
| 9660: | 20 | 31 | 35 | 20 | 31 | 38 | 20 | 32 | 32 | 20 | 32 | 36 | 20 | 33 | 31 | 20 | |
| 9670: | 33 | 37 | 20 | 34 | 34 | 20 | 35 | 33 | 20 | 36 | 33 | 20 | 37 | 34 | 20 | 38 | |
| 9680: | 37 | 31 | 30 | 32 | 31 | 31 | 38 | 31 | 33 | 36 | 31 | 35 | 36 | 31 | 37 | 39 | |
| 9690: | 32 | 30 | 35 | 32 | 33 | 34 | 32 | 36 | 37 | 33 | 30 | 34 | 33 | 34 | 35 | 33 | |
| 96A0: | 39 | 38 | 34 | 35 | 31 | 35 | 31 | 38 | 35 | 37 | 33 | 36 | 32 | 35 | 36 | 39 | |
| 96B0: | 31 | 37 | 35 | 30 | 01 | 01 | 01 | 01 | 00 | 02 | 03 | 03 | 00 | 00 | 01 | 02 | |
| 96C0: | 04 | 01 | 00 | 02 | 05 | 06 | 01 | 00 | 01 | 01 | 01 | 01 | 01 | 02 | 04 | 03 | |
| 96D0: | 04 | 00 | 03 | 05 | 06 | 06 | 00 | 01 | 01 | 01 | 00 | 00 | 04 | 04 | 02 | 07 | |
| 96E0: | 00 | 01 | 02 | 01 | 03 | 04 | 04 | 02 | 06 | 01 | 01 | 01 | 01 | 01 | 01 | 00 | |
| 96F0: | 04 | 04 | 04 | 00 | 00 | 03 | 05 | 06 | 02 | 01 | 07 | 07 | 07 | 00 | 00 | 01 | |
| 9700: | 01 | 01 | 01 | 00 | 03 | 03 | 03 | 03 | 03 | 01 | 01 | 01 | 01 | 01 | 06 | 05 | |
| 9710: | 04 | 03 | 02 | 07 | 07 | 04 | 05 | 00 | 04 | 03 | 07 | 06 | 01 | 02 | 01 | 06 | |
| 9720: | 07 | 04 | 07 | 07 | 07 | 07 | 00 | 02 | 03 | 04 | 04 | 04 | 04 | 04 | 05 | 07 | |
| 9730: | 05 | 02 | 06 | 06 | 07 | 04 | 03 | 01 | 01 | 07 | 07 | 07 | 07 | 07 | 07 | 01 | |
| 9740: | 03 | 04 | 07 | 07 | 07 | 04 | 07 | 07 | 04 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | |
| 9750: | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | |
| 9760: | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | |
| 9770: | 36 | 34 | 00 | 00 | 32 | 32 | 32 | 32 | 00 | 2F | 34 | 32 | 30 | 00 | 35 | 35 | |
| 9780: | 35 | 35 | 35 | 2B | 33 | 2E | 33 | 00 | 33 | 32 | 32 | 35 | 00 | 25 | 25 | 25 | |
| 9790: | 00 | 00 | 31 | 31 | 2C | 34 | 00 | 32 | 34 | 33 | 35 | 33 | 32 | 2D | 32 | 35 | |
| 97A0: | 30 | 27 | 27 | 27 | 27 | 00 | 2D | 2D | 2D | 00 | 00 | 2B | 30 | 2D | 30 | 29 | |
| 97B0: | 30 | 30 | 30 | 00 | 00 | 23 | 29 | 25 | 25 | 00 | 2C | 2C | 2C | 2D | 2A | 25 | |
| 97C0: | 26 | 27 | 26 | 25 | 2C | 2C | 30 | 2B | 28 | 2E | 31 | 2E | 2B | 00 | 2E | 2B | |
| 97D0: | 2D | 2A | 24 | 25 | 22 | 2B | 30 | 02 | 29 | 2B | 2D | 29 | 00 | 30 | 2D | 30 | |
| 97E0: | 2A | 27 | 27 | 2B | 2B | 30 | 30 | 2A | 2C | 25 | 26 | 2D | 2B | 2A | 29 | 2B | |
| 97F0: | 23 | 28 | 2B | 2A | 2B | 29 | 2A | 2D | 2D | 2A | 2C | 27 | 27 | 27 | 27 | 27 | |
| 9800: | 03 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 03 | 01 | 01 | 01 | |
| 9810: | 01 | 01 | 01 | 01 | 03 | 01 | 01 | 01 | 01 | 00 | A0 | 00 | 01 | 01 | 01 | 01 | |
| 9820: | 03 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 03 | 01 | 01 | 01 | 01 | 01 | 01 | |
| 9830: | 01 | 01 | 01 | 01 | 03 | 00 | 03 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 03 | |
| 9840: | 03 | 00 | 35 | 5D | 55 | 5D | 55 | 5D | 55 | 5D | 65 | 00 | 03 | 00 | 34 | 54 | |
| 9850: | 54 | 54 | 6A | 00 | 03 | 00 | 34 | 54 | 54 | D4 | 54 | 54 | 54 | 54 | 64 | 00 | |
| 9860: | 03 | 00 | 3B | 54 | 54 | 54 | 64 | 00 | 03 | 00 | 35 | 5D | 55 | 5D | 55 | 5D | |
| 9870: | 55 | 5D | 65 | 00 | 03 | 00 | 03 | 00 | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 03 | |
| 9880: | 03 | 00 | AD | 00 | 02 | 02 | 02 | 02 | 02 | 00 | A0 | 00 | 03 | 00 | A4 | 00 | |
| 9890: | 02 | 00 | A4 | 00 | 03 | 00 | A4 | 00 | 02 | 02 | 02 | 02 | 02 | 00 | A4 | 00 | |
| 98A0: | 03 | 00 | A4 | 00 | 02 | 00 | A4 | 00 | 03 | 00 | A0 | 00 | 02 | 02 | 02 | 02 | |
| 98B0: | 02 | 00 | AD | 00 | 03 | 00 | 03 | 00 | 0F | 0F | 0F | 0F | 0F | 0F | 00 | 03 | |
| 98C0: | 03 | 00 | A5 | 00 | 03 | 01 | 01 | 01 | 01 | 08 | A8 | 08 | 01 | 00 | A4 | 00 | |
| 98D0: | 03 | 00 | A4 | 00 | 01 | 00 | A4 | 00 | 01 | 01 | 01 | 01 | 01 | 00 | A4 | 00 | |
| 98E0: | 01 | 00 | A4 | 00 | 03 | 00 | A4 | 00 | 01 | 08 | A8 | 08 | 01 | 01 | 01 | 01 | |
| 98F0: | 03 | 00 | A5 | 00 | 03 | 00 | 03 | 00 | 0F | 0F | 0F | 0F | 0F | 0F | 00 | 03 | |
| 9900: | 03 | 00 | AD | 00 | 03 | 00 | 37 | 54 | 54 | 54 | D4 | 54 | 54 | 54 | E4 | 00 | |
| 9910: | 03 | 00 | B4 | 54 | 54 | 54 | D4 | 54 | 54 | 54 | 74 | 54 | 54 | 54 | D4 | 54 | |
| 9920: | 54 | 54 | E4 | 00 | 03 | 00 | B4 | 54 | 54 | 54 | D4 | 54 | 54 | 54 | 67 | 00 | |
| 9930: | 03 | 00 | AD | 00 | 03 | 00 | 03 | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 03 | |
| 9940: | 03 | 00 | A5 | 00 | 03 | 00 | A4 | 00 | 02 | 02 | 02 | 02 | 02 | 00 | A4 | 00 | |
| 9950: | 03 | 00 | A4 | 00 | 02 | 02 | 02 | 02 | 02 | 00 | A4 | 00 | 02 | 02 | 02 | 02 | |
| 9960: | 02 | 00 | A4 | 00 | 03 | 00 | A4 | 00 | 02 | 02 | 02 | 02 | 02 | 00 | A4 | 00 | |
| 9970: | 03 | 00 | A5 | 00 | 03 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 9980: | 03 | 08 | A8 | 08 | 01 | 00 | A4 | 00 | 01 | 01 | 01 | 01 | 03 | 00 | A4 | 00 | |
| 9990: | 01 | 00 | A4 | 00 | 01 | 01 | 01 | 01 | 01 | 00 | A4 | 00 | 01 | 01 | 01 | 01 | |
| 99A0: | 01 | 00 | A4 | 00 | 01 | 00 | A4 | 00 | 03 | 01 | 01 | 01 | 01 | 01 | 00 | A4 | 00 |
| 99B0: | 01 | 88 | A8 | 08 | 03 | 00 | 03 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 03 | |
| 99C0: | 03 | 00 | B4 | 54 | 54 | 54 | D4 | 54 | 54 | 54 | 64 | 00 | 03 | 00 | 94 | 54 | |
| 99D0: | 54 | 54 | F4 | 54 | 54 | 54 | 54 | 54 | 54 | F4 | 54 | 54 | 54 | 54 | 54 | 54 | |
| 99E0: | 54 | 54 | F4 | 54 | 54 | 54 | C4 | 00 | 03 | 00 | 34 | 54 | 54 | 54 | D4 | 54 | |
| 99F0: | 54 | 54 | E4 | 00 | 03 | 00 | 03 | 00 | 0F | 0F | 0F | 0F | 0F | 0F | 00 | 03 | |
| 9A00: | | | | | | | | | | | | | | | | | |

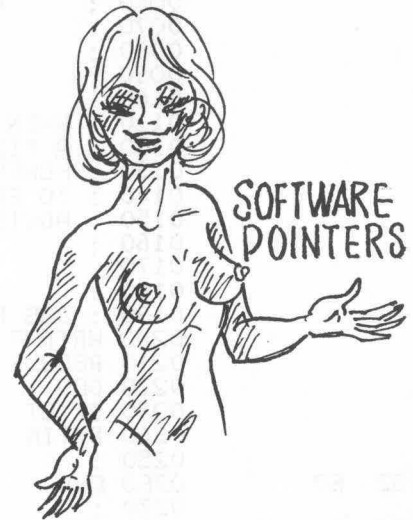
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 9A00: | 03 | 00 | A4 | 00 | 02 | 02 | 02 | 02 | 02 | 00 | A4 | 00 | 03 | 02 | 02 | 02 | |
| 9A10: | 02 | 09 | A9 | 09 | 02 | 02 | 02 | 02 | 02 | 00 | A4 | 00 | 02 | 02 | 02 | 02 | |
| 9A20: | 02 | 09 | A9 | 09 | 02 | 02 | 02 | 02 | 02 | 03 | 00 | A4 | 00 | 02 | 02 | 02 | |
| 9A30: | 02 | 00 | A4 | 00 | 03 | 00 | 03 | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 03 | |
| 9A40: | 03 | 00 | A4 | 00 | 01 | 01 | 01 | 01 | 01 | 00 | A4 | 00 | 03 | 01 | 01 | 01 | |
| 9A50: | 01 | 00 | A5 | 00 | 03 | 01 | 01 | 01 | 01 | 03 | 00 | A4 | 00 | 03 | 01 | 01 | |
| 9A60: | 03 | 00 | A5 | 00 | 01 | 01 | 01 | 01 | 01 | 03 | 00 | A4 | 00 | 01 | 01 | 01 | |
| 9A70: | 01 | 00 | A4 | 00 | 03 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 9A80: | 03 | 00 | 9B | 54 | 54 | 54 | 74 | 54 | 54 | 54 | E4 | 00 | 03 | 00 | 3D | 55 | |
| 9A90: | 5D | 55 | CD | 00 | 03 | 00 | 0F | 00 | 03 | 00 | A4 | 00 | 03 | 00 | 00 | 00 | |
| 9AA0: | 03 | 00 | 9D | 55 | 5D | 55 | 6D | 00 | 03 | 00 | B4 | 54 | 54 | 54 | 74 | 54 | |
| 9AB0: | 54 | 54 | CA | 00 | 03 | 00 | 03 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 03 | |
| 9AC0: | 03 | 02 | 02 | 02 | 02 | 00 | A4 | 00 | 02 | 00 | A4 | 00 | 03 | 00 | A5 | 00 | |
| 9AD0: | 02 | 02 | 02 | 02 | 03 | 00 | 0F | 00 | 03 | 00 | A4 | 00 | 03 | 00 | 00 | 00 | |
| 9AE0: | 03 | 02 | 02 | 02 | 02 | 00 | A5 | 00 | 03 | 00 | A4 | 00 | 02 | 00 | A4 | 00 | |
| 9AF0: | 02 | 02 | 02 | 02 | 03 | 00 | 03 | 00 | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 03 | |
| 9B00: | 01 | 01 | 01 | 01 | 01 | 00 | A4 | 00 | 03 | 00 | A4 | 00 | 01 | 08 | A8 | 08 | |
| 9B10: | 01 | 01 | 01 | 01 | 01 | 00 | 0F | 00 | 01 | 00 | A4 | 00 | 01 | 00 | 00 | 00 | |
| 9B20: | 01 | 01 | 01 | 01 | 01 | 01 | 08 | A8 | 08 | 01 | 00 | A4 | 00 | 03 | 00 | A4 | 00 |
| 9B30: | 01 | 01 | 01 | 01 | 01 | 00 | 03 | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 03 | |
| 9B40: | 50 | 50 | 50 | 50 | 50 | 50 | E4 | 00 | 03 | 00 | B4 | 54 | 54 | 54 | F4 | 50 | |
| 9B50: | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | F0 | 50 | 50 | 50 | 50 | 50 | |
| 9B60: | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | F4 | 54 | 54 | 54 | E4 | 00 | |
| 9B70: | 50 | 50 | 50 | 50 | 50 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 9B80: | 02 | 02 | 02 | 02 | 02 | 00 | A4 | 00 | 03 | 00 | A4 | 00 | 02 | 09 | A9 | 09 | |
| 9B90: | 02 | 02 | 02 | 02 | 02 | 00 | 0F | 00 | 02 | 00 | A4 | 00 | 02 | 00 | 00 | 00 | |
| 9BA0: | 02 | 02 | 02 | 02 | 02 | 09 | A9 | 09 | 02 | 00 | A4 | 00 | 03 | 00 | A4 | 00 | |
| 9BB0: | 02 | 02 | 02 | 02 | 02 | 00 | 0C | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 0C | |
| 9BC0: | 03 | 01 | 01 | 01 | 01 | 00 | A4 | 00 | 01 | 00 | A4 | 00 | 03 | 00 | A5 | 00 | |
| 9BD0: | 01 | 01 | 01 | 01 | 03 | 00 | 0F | 00 | 03 | 00 | A4 | 00 | 03 | 00 | 00 | 00 | |
| 9BE0: | 03 | 01 | 01 | 01 | 01 | 00 | A5 | 00 | 03 | 00 | A4 | 00 | 01 | 00 | A4 | 00 | |
| 9BF0: | 01 | 01 | 01 | 01 | 03 | 00 | 0C | 00 | 0 | | | | | | | | |

```

0 1 2 3 4 5 6 7 8 9 A B C D E F
9E00: 03 00 AD 00 01 01 01 01 01 00 A0 00 03 00 A4 00
9E10: 01 00 A4 00 03 00 A4 00 01 01 01 01 01 00 A4 00
9E20: 03 00 A4 00 01 00 A4 00 03 00 A0 00 01 01 01 01
9E30: 01 00 AD 00 03 00 03 01 01 01 01 01 01 01 03
9E40: 03 00 95 5D 55 5D 55 5D 55 5D C5 00 03 00 94 54
9E50: 54 54 CA 00 03 00 94 54 54 54 74 54 54 54 C4 00
9E60: 03 00 9B 54 54 54 C4 00 03 00 95 5D 55 5D 55 5D
9E70: 55 5D C5 00 03 00 03 00 0F 0F 0F 0F 0F 0F 00 03
9E80: 03 02 02 02 02 02 02 02 02 02 02 02 03 02 02 02
9E90: 02 02 02 02 03 02 02 02 02 00 A0 00 02 02 02 02
9EA0: 03 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
9EB0: 02 02 02 02 03 00 03 00 00 00 00 00 00 00 00 03
9EC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9ED0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9EE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9EF0: 00 00 00 00 00 00 03 00 0F 0F 0F 0F 0F 0F 00 03
9F00: 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C
9F10: 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C
9F20: 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C
9F30: 0C 0C 0C 0C 0C 00 03 02 02 02 02 02 02 02 02 03
9F40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9F50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9F60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9F70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9F80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9F90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9FA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9FB0: 00 00 00 00 00 00 00 08 00 00 00 00 00 00 00 00
9FC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9FD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9FE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9FF0: 00 00 00 08 00 00 00 00 00 00 00 00 00 00 00 00
A000:

```

JUNIOR



SOFTWARE
POINTERS

Is Uw computer niet uitgevoerd met een machinetaal monitor dan kan men met PEEK en POKE even snel een geheugenplaats bekijken of wijzigen. Wil men meerdere plaatsen bekijken dan wordt dit toch een tijdrovende zaak. Toets echter onderstaand korte programma in, en U heeft een direkt overzicht van 40 adressen. Het aantal te printen adressen is te wijzigen in regel 40. Het startadres dient decimaal ingeseven te worden.

```

10 INPUT"GEEF STARTADRES":A
20 PRINTA"--PEEK(A),
30 A=A+1:B=B+1
40 IFB=40THENEND
50 GOTO20

```

GEEF STARTADRES? 49152

| | | | | |
|-------------|-------------|-------------|-------------|-------------|
| 49152 - 76 | 49153 - 109 | 49154 - 222 | 49155 - 35 | 49156 - 198 |
| 49157 - 52 | 49158 - 197 | 49159 - 215 | 49160 - 202 | 49161 - 129 |
| 49162 - 199 | 49163 - 184 | 49164 - 201 | 49165 - 84 | 49166 - 206 |
| 49167 - 228 | 49168 - 201 | 49169 - 46 | 49170 - 200 | 49171 - 46 |
| 49172 - 199 | 49173 - 6 | 49174 - 199 | 49175 - 177 | 49176 - 199 |
| 49177 - 9 | 49178 - 198 | 49179 - 17 | 49180 - 199 | 49181 - 91 |
| 49182 - 199 | 49183 - 196 | 49184 - 199 | 49185 - 33 | 49186 - 198 |
| 49187 - 212 | 49188 - 199 | 49189 - 100 | 49190 - 198 | 49191 - 226 |

```

0001      .LS
0010 : *** BUG IN DISC PATCHES ON Carl MOSER'S ***
0020 : *** ASSEMBLER ASSM/TED FOR Disc Operating *
0030 : *** Svsstem DOSAV2.0A BY Koen van Nieuwen- *
0040 : *** hove. Asse. Belgium ***
0050 :
0060 :     AUTHOR: HANS BUURMAN
0070 :     VAN DER FUYCKSTRAAT 136
0080 :     3232 AS BRIELLE
0090 :     THE NETHERLANDS
0100 :
0110 : WHEN ON THE OLD PATCH A 'PU X' COMMAND WAS GIVEN.
0120 : A FILE WITH LENGTH 0 SHOULD HAVE BEEN WRITTEN.
0130 : HOWEVER, A 64K FILE WAS WRITTEN.
0140 : TO ECONOMISE ON DISC SPACE, FOLLOWING CHANGE IS
0150 : ADVISED:
0160 :
0170 :     .BA $4162
0180 :
0190 : DOS REFERENCES:
0200 WRTBUF      .DE $458
0210 REDBUF      .DE $3D0
0220 DOS          .DE $7C00 (IN MY DOS)
0230 BYTOT       .DE DOS+$24
0240 BYTIN        .DE DOS+$27
0250 :
4162- 60      0260 END      RTS
0270 :
4163- 20 87 41 0280 DISCO.VEC JSR INIT
4166- A5 3E      0290 NEXT   LDA #$3E
4168- CD 26 01   0300      CMP $126 ($127,$126) CONTAINS END OF FILE
416B- D0 07      0310      BNE NEXT1
416D- A5 3F      0320      LDA #$3F
416F- CD 27 01   0330      CMP $127
4172- F0 EE      0340      BEQ END
4174- A0 00      0350 NEXT1   LDY #0
4176- B1 3E      0360      LDA ($3E),Y
4178- A2 58      0370      LDX #L.WRTBUF
417A- A0 04      0380      LDY #H.WRTBUF
417C- 20 24 7C   0390      JSR BYTOT WRITE BYTE TO DISC
417F- E6 3E      0400      INC #$3E
4181- D0 E3      0410      BNE NEXT
4183- E6 3F      0420      INC #$3F
4185- D0 DF      0430      BNE NEXT
4187- AD 24 01   0440 INIT    LDA $124 ($125,$124) CONTAINS STARTADDRESS
418A- 85 3E      0450      STA #$3E
418C- AD 25 01   0460      LDA $125
418F- 85 3F      0470      STA #$3F
4191- 60      0480 NEXT4   RTS
0490 :
4192- 20 87 41 0500 DISCI.VEC JSR INIT
4195- A5 3E      0510 NEXT3   LDA #$3E
4197- CD 26 01   0520      CMP $126 ($127,$126) CONTAINS END-ADDRESS
419A- D0 07      0530      BNE NEXT2
419C- A5 3F      0540      LDA #$3F
419E- CD 27 01   0550      CMP $127
41A1- F0 EE      0560      BEQ NEXT4
41A3- A2 D0      0570 NEXT2   LDX #L.REDBUF
41A5- A0 03      0580      LDY #H.REDBUF
41A7- 20 27 7C   0590      JSR BYTIN READ BYTE FROM DISK
41AA- 48      0600      PHA
41AB- AD 23 01   0610      LDA $123
41AE- F0 07      0620      BEQ NEXT5 DO NOT STORE IF $123=0
41B0- 68      0630      PLA
41B1- A0 00      0640      LDY #0 ELSE STORE BYTE
41B3- 91 3E      0650      STA ($3E),Y
41B5- F0 01      0660      BEQ NEXT6
41B7- 68      0670 NEXT5   PLA
41B8- E6 3E      0680 NEXT6   INC #$3E
41BA- D0 D9      0690      BNE NEXT3
41BC- E6 3F      0700      INC #$3F
41BE- D0 D5      0710      BNE NEXT3
0720 :
0730 :     .BA $41D9
41D9- A9 92      0740      LDA #L.DISCI.VEC
0750 :
//0000,41DB,41DB

```

PROTON UTILITY SOURCE

Onder de titel " Proton utility source " zal ik in een aantal 6502 kenners, de 5 Proton hulpprogramma's < *COPY* > < MAP* > < OLOAD* > < UTIL* > < *COMP* > publiceren.

Deze source worden met toestemming van Proton gepubliceerd.

De aanleiding van deze publicatie is een aantal kleine foutjes in de code-file's, < *COMP* > < UTIL* > < *COPY* > .

De firma Proton heeft deze kleine foutjes op mijn verzoek terstond verbeterd en gelijktijdig toestemming tot publicatie gegeven.

Ik denk dat dit voor de Proton bezitters prettig nieuws is.

Een ieder kan de source zelf intoetsen en daarna assembleren en dan de nieuwe verbeterde code gaan gebruiken.

Het door Proton geleverde commentaar is beperkt, maar voor een Proton gebruiker volgens mij voldoende.

Mocht er echter behoefte bestaan aan veel meer commentaar laat het dan even aan mij weten, dan zal ik bij de volgende file's zelf meer commentaar toevoegen.

Het eerste programma dat ik heb uitgekozen is de offset loader.

Alle utility programma's zijn bedoeld voor Disk-functie's, zodat het voor het begrijpen van de source file's makkelijk is als we weten hoe Proton de code op de floppy's wegschrijft.

Met behulp van een FORTH programma wat geschreven is in Proton FORTH, kan ik elke gewenste sector van af een floppy uitprinten.

Ik heb dan ook een dump van de code van het programma OLOAD* hier bij afgedrukt.

Uit deze dump is ook duidelijk te zien dat de code totaal 4 sectoren van de floppy gebruikt, zoals ook uit een opgave van de directory blijkt. In dit voorbeeld stond de code bij mij op track 32 sector 5 tot sector 8. We zien dat de eerste byte van het programma een een 0D <CR> is en dat de volgende byte de ascii waarde 3B t.w. <;> is.

Deze 3B wordt gebruikt als regel scheider.

Elke nieuwe code regel begint met 3B.

Direct achter deze 3B staat het aantal byte's dat in deze regel is weggeschreven in ons geval <02>.

De volgende 2 byte's bevatten het memory-adres waar de code weggeschreven moet worden. De volgorde is <H> <L>, hier 05 06, t.w. \$506.

Dan komen de code byte's in dit geval dus 2, na de code byte's volgt de checksum, bestaande uit 2 byte's. In ons geval 000F. En dat klopt want de hexadecimale som van alle byte's is 02 + 05 + 06 + 00 + 02 = 0F. Vervolgens komt er weer een 0D die weer voor <CR> staat.

In ons voorbeeld zien we dat er dan nog een 0D volgt, dit heeft geen gevolgen want alle loadprogramma's bij Proton gaan nu op zoek naar de regelscheider 3B <;> .

In de volgende code regel zien we dat er 18 byte's zijn weggeschreven en dat het start adres \$200 is.

De checksum is in dit geval 592, wat uiteraard juist is.

De volgende sectoren spreken voor zich zelf.

Op het moment dat er 0 byte's als aantal op de floppy worden gelezen stopt het programma met inlezen.

Ik hoop dat deze uitleg meer inzicht geeft in de werking hoe de code op floppy wordt weggeschreven.

Men moet dit format niet hanteren bij sourcefile's die met de editor worden weggeschreven dat gaat anders, waarover later meer.

Het FORHT programma dat ik voor het dumpen van de sectoren heb gebruikt zal ik na de publicatie van de sourcefile's eveneens publiceren.

```
<5>
PD5-65 FORTH V1.0
SEC
```

Piet Fransen
Voordewind 21
Amsterdam.
tel. 020 319271

```
Floppy nr : 2
File name : OLOAD D*
BEGIN AT TRACK = 32      SECTOR = 5
END AT TRACK = 32      SECTOR = 8
NUMBER OF SECTORS 4
OK
NAME-DUMP
```

*** Floppy dumper ***

```
Floppy nr : 2
File name : OLOAD*
Output to Printer or Terminal ( P/T )
```

| TRACK= 32 | SECTOR= 5 | |
|-----------|---|-------------------------------|
| 8000 | 0D 3B 02 05 06 00 02 00 0F 0D 0D 3B 18 02 00 D8 | . ; ; ... X |
| 8010 | 20 25 03 A2 00 20 AB 02 20 25 03 20 14 03 20 25 | . % . " . + . % . . . % |
| 8020 | 03 A2 13 20 AB 02 A0 05 92 0D 3B 18 02 18 13 20 | . " . + ; |
| 8030 | F2 02 20 25 03 20 29 02 20 1F 03 20 25 03 60 20 | r . % . .) % . ' . |
| 8040 | 34 02 90 FB A2 34 05 8D 0D 3B 18 02 30 20 AB 02 | 4 . { " 4 . . ; . . 0 . + . |
| 8050 | 60 20 17 03 C9 3B D0 F9 A9 00 8D 24 05 8D 25 05 | ' . . . I ; Py) . . \$. % . |
| 8060 | 20 2E 03 20 34 07 39 0D 3B 18 02 48 03 85 10 F0 | 4 . 9 . ; . . H . . p |
| 8070 | 47 20 2E 03 20 34 03 85 12 20 2E 03 20 34 03 85 | G 4 4 . . |

| TRACK= 32 | SECTOR= 6 | |
|-----------|---|----------------------------------|
| 8000 | 11 20 9A 02 05 6A 0D 3B 18 02 60 A0 00 20 2E 03 | j . ; . . ' |
| 8010 | 91 15 D1 15 D0 1A 20 34 03 C8 C6 10 D0 EF 20 2E | . . 0 . P . . 4 . HF . Po . . |
| 8020 | 03 CD 25 06 D8 0D 3B 18 02 78 05 D0 1B 20 2E 03 | . M % . X . ; . . x . P |
| 8030 | CD 24 05 D0 13 18 60 A2 1A 20 AB 02 A5 16 A6 15 | M \$. P . . ' " . . + . % . & . |
| 8040 | 20 2B 07 6E 0D 3B 18 02 90 03 20 25 03 38 60 A2 | . + . n . ; % . 8 ' " |
| 8050 | 24 D0 ED 18 A5 11 65 13 85 15 A5 12 65 14 85 16 | \$ P m . % . e . . . % . e . . . |
| 8060 | 60 08 1B 0D 3B 18 02 A6 20 1C 03 BD B4 02 E8 C9 | ' . . . ; . . (. . . = 4 . h I |
| 8070 | 3B D0 F5 60 4F 46 46 53 45 54 20 4C 4F 41 44 45 | ; P u ' OFF SET . LOADE |

TRACK= 32

SECTOR= 7

```
8000 09 D1 0D 3B 18 02 C0 52 20 56 31 2E 30 3B 4F 46
8010 46 53 45 54 3B 4D 45 4D 2D 46 41 49 4C 20 3B 06
8020 F1 0D 3B 18 02 D8 43 48 45 43 4B 53 55 4D 20 45
8030 52 52 4F 52 20 3B 2A 2A 20 45 4E 44 20 2A 06 DF
8040 0D 3B 18 02 F0 2A 3B 98 48 20 31 03 68 A8 AD 24
8050 05 D0 0D AD 22 05 99 00 00 AD 23 05 99 08 41 0D
8060 3B 18 03 08 01 00 60 A9 00 99 00 00 99 01 00 60
8070 6C 02 E0 6C 06 E0 A9 20 6C 08 E0 6C 07 E9 0D 3B
```

```
.Q.;..OR.V1.0;OF
FSET;MEM-FAIL.;.
Q.;..XCHECKSUM.E
RROR.;**.*.END.*._
.;..p*.;.H.1.h(-$
.P.-".....-#...A.
;.....').....'
1.'1.'1.'1.'1.'1.;
```

TRACK= 32

SECTOR= 8

```
8000 17 03 20 0A E0 6C 24 E0 6C 0E E0 6C 14 E0 6C 26
8010 E0 6C 22 E0 6C 48 E0 6C 2C E0 0B 3A 0D 3B 00 00
8020 0F 00 0F 00 00 00 00 00 00 00 00 00 00 00 00
8030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
.....'1$'1.'1.'1.&
'1''1H'1,'.3.;..
.....
.....
.....
.....
.....
.....
.....
```

OK

PASS 1

PASS 2

OLOAD-UTILITY

```
0001 0000 .TITLE ' #### OLOAD-UTILITY ####'
0002 0000 ; FILE = OLOAD1 <<<< FOR JUNIOR/KIM >>>>
0003 0000 ; DD: 27-11-82 SK
0004 0000 ;
0005 0000 ; *****
0006 0000 ; *****
0007 0000 ; *****
0008 0000 ; ***** SHOW THE CODE-LAY-OUT *****
0009 0000 ; ***** AND LOAD WITH OFFSET. *****
0010 0000 ; *****
0011 0000 ; *****
0012 0000 ; *****
0013 0000 ;
0014 0000 ; ### MONITOR LINK ###
0015 0000 ;
0016 0000 ENTR = $E000
0017 0000 MONRAM = $0500
0018 0000 ;
0019 0000 ADDR = MONRAM+34
0020 0000 CKSUM = MONRAM+36
0021 0000 ;
0022 0000 ; *= MONRAM+6
0023 0506 0002 .WOR OLOAD
0024 0508 ;
0025 0508 ;
0026 0508 ; *= $10
0027 0010 ; ### VARIABLES ###
0028 0010 ;
0029 0010 LENGTH ; ==+1 ; LENGTH OF INPUT-LINE
0030 0011 ADDRES ; ==+2 ; ADDRES OF LINE
0031 0013 OFFSET ; ==+2 ; OFFSET ON THE ADDRES
0032 0015 MEMPTR ; ==+2 ; POINTER TO THE MEMORY
0033 0017 YSAVE ; ==+1
```

```

0034 0018 ;
0035 0018 ;
0036 0018 ;
0037 0018 **=$0200
0038 0200 ;
0039 0200 ; ## MAIN PROGRAM BODY ##
0040 0200 ;
0041 0200 D8 OLOAD: CLD
0042 0201 202503 JSR CRLF
0043 0204 A200 LDX #TID ; "OFFSET-LOADER VX.X"
0044 0206 20AB02 JSR PRINT
0045 0209 202503 JSR CRLF
0046 020C 201403 JSR WHEREI ; INPUT-DEVICE
0047 020F 202503 JSR CRLF
0048 0212 A213 LDX #TOFFST
0049 0214 20AB02 JSR PRINT
0050 0217 A013 LDY #OFFSET
0051 0219 20F202 JSR GETREG
0052 021C 202503 JSR CRLF
0053 021F ;
0054 021F 202902 JSR OLOAD1 ; LOAD THE FILE
0055 0222 201F03 JSR CLOSEI ; CLOSE THE INPUTFILE
0056 0225 202503 JSR CRLF
0057 0228 60 RTS
0058 0229 ;
0059 0229 ;
0060 0229 ; ## OLOAD-ROUTINE ##
0061 0229 ;
0062 0229 203402 OLOAD1 JSR RADDR ; READ LINE, ADDRESS & LENGTH
0063 022C 90FB BCC OLOAD1 ; *EOF* ?
0064 022E A234 LDX #TEND
0065 0230 20AB02 JSR PRINT
0066 0233 60 RTS
0067 0234 ;
0068 0234 ;
0069 0234 ; ## READ LINE, ADDRESS & LENGTH ##
0070 0234 ;
0071 0234 201703 RADDR JSR INALL ; SCAN TILL ';'
0072 0237 C93B CMP #' ;
0073 0239 D0F9 BNE RADDR
0074 023B A900 LDA #0 ; CLEAR THE CHECKSUM
0075 023D 8D2405 STA CKSUM
0076 0240 8D2505 STA CKSUM+1
0077 0243 202E03 JSR RBYTE ; LENGTH
0078 0246 203403 JSR CHEKA
0079 0249 8510 STA LENGTH
0080 024B F047 BEQ RADDR9 ; *EOF*
0081 024D 202E03 JSR RBYTE
0082 0250 203403 JSR CHEKA
0083 0253 8512 STA ADDRESS+1
0084 0255 202E03 JSR RBYTE
0085 0258 203403 JSR CHEKA
0086 025B 8511 STA ADDRESS
0087 025D 209A02 JSR CALOFF ; CALCULATE OFFSET
0088 0260 A000 LDY #0
0089 0262 ; NOW READ THE LINE
0090 0262 202E03 RADDR1 JSR RBYTE
0091 0265 9115 STA (MEMPTR)Y SAVE THE BYTE
0092 0267 D115 CMP (MEMPTR)Y ; AND CHECK FOR RAM

```

```

0093 0269 D01A          BNE MERR
0094 026B 203403        JSR CHEKA      ; ADD TO CHECKSUM
0095 026E C8           INV
0096 026F C610          DEC LENGTH
0097 0271 D0EF          BNE RADDR1
0098 0273 202E03        JSR RBYTE
0099 0276 CD2505        CMP CKSUM+1
0100 0279 D01B          BNE CKERR
0101 027B 202E03        JSR RBYTE
0102 027E CD2405        CMP CKSUM
0103 0281 D013          BNE CKERR
0104 0283 18           CLC
0105 0284 60           RTS
0106 0285              ;
0107 0285 A21A MERR      LDX #TMERR
0108 0287 20AB02 ERROR   JSR PRINT
0109 028A A516          LDA MEMPTR+1
0110 028C A615          LDX MEMPTR
0111 028E 202B03        JSR WRAX
0112 0291 202503        JSR CRLF
0113 0294              ;
0114 0294 38 RADDR9     SEC
0115 0295 60           RTS
0116 0296              ;
0117 0296 A224 CKERR    LDX #TCKERR
0118 0298 D0ED          BNE ERROR
0119 029A              ;
0120 029A              ;
0121 029A              ; ### CALCULATE MEMPTR=ADDRESS+OFFSET ###
0122 029A              ;
0123 029A 18 CALOFF     CLC
0124 029B A511          LDA ADDRESS
0125 029D 6513          ADC OFFSET
0126 029F 8515          STA MEMPTR
0127 02A1 A512          LDA ADDRESS+1
0128 02A3 6514          ADC OFFSET+1
0129 02A5 8516          STA MEMPTR+1
0130 02A7 60           RTS
0131 02A8              ;
0132 02A8              ;
0133 02A8              ;
0134 02A8              ; ## PRINT STRING, INDEX BY .X ##
0135 02A8              ;
0136 02A8 201C03 PRNT1   JSR OUTALL
0137 02AB BDB402 PRINT   LDA TXT,X
0138 02AE E8           INX
0139 02AF C93B          CMP #' ;
0140 02B1 D0F5          BNE PRNT1
0141 02B3 60           RTS
0142 02B4              ;
0143 02B4              ;
0144 02B4              ; ## TEXT-TABLE ##
0145 02B4              ;
0146 02B4 TXT           ==:
0147 02B4              ;
0148 02B4 TID           ==:-TXT
0149 02B4 4F46          .BYT 'OFFSET LOADER V1.0; '
0150 02C7 TOFFST       ==:-TXT

```

```

0151 02C7 4F46 .BYT 'OFFSET;'
0152 02CE TMERR ==*-TXT
0153 02CE 4D45 .BYT 'MEM-FAIL ;'
0154 02D8 TCKERR ==*-TXT
0155 02D8 4348 .BYT 'CHECKSUM ERROR ;'
0156 02E8 TEND ==*-TXT
0157 02E8 2A2A .BYT '** END **;'
0158 02F2 ;
0159 02F2 98 GETREG TYA
0160 02F3 48 PHA
0161 02F4 203103 JSR ADDIN ; GET ADDRESS VALUE
0162 02F7 68 PLA
0163 02F8 A8 TAY
0164 02F9 AD2405 LDA CKSUM
0165 02FC D00D BNE DEFLT
0166 02FE AD2205 LDA ADDR
0167 0301 990000 STA 0,Y
0168 0304 AD2305 LDA ADDR+1
0169 0307 990100 STA 1,Y
0170 030A 60 RTS
0171 030B ;
0172 030B A900 DEFLT LDA #0
0173 030D 990000 STA 0,Y
0174 0310 990100 STA 1,Y
0175 0313 60 RTS
0176 0314 ;
0177 0314 ;
0178 0314 ;
0179 0314 ; ##### MONITOR ENTRIES #####
0180 0314 ;
0181 0314 6C02E0 WHEREI JMP (ENTR+2)
0182 0317 6C06E0 INALL JMP (ENTR+6)
0183 031A A920 BLANK LDA #$20
0184 031C 6C08E0 OUTALL JMP (ENTR+8)
0185 031F 6C0AE0 CLOSEI JMP (ENTR+10)
0186 0322 6C24E0 NUMA JMP (ENTR+36)
0187 0325 6C0EE0 CRLF JMP (ENTR+14)
0188 0328 6C14E0 RCHEK JMP (ENTR+20)
0189 032B 6C26E0 WRAX JMP (ENTR+38)
0190 032E 6C22E0 RBYTE JMP (ENTR+34)
0191 0331 6C48E0 ADDIN JMP (ENTR+72)
0192 0334 6C2CE0 CHEKA JMP (ENTR+44)
0193 0337 ;
0194 0337 ;
0195 0337 .END

```

ERRORS: 0000

<0000>

AANPASSING OS-65D V3.3 PRINT&(X,Y) COMMANDO =====

De hieronder beschreven aanpassing is bestemd voor de bezitters van een JUNIOR computer met Elektuur VDU-kaart en het OHIO SCIENTIFIC OS-65D V3.3 disk operating system.

De bij het disk operating system behorende BASIC versie (OSI NINE DIGIT BASIC) kent een aantal zogenaamde CURSOR CONTROL commando's die in verband met de afwijkende hardware niet werken bij de JUNIOR computer. Een van die commando's is het PRINT&(X,Y) commando waarmee de gebruiker in staat wordt gesteld om de cursor op het beeldscherm te verplaatsen naar de coördinaten zoals opgegeven in de X en Y variabelen. Met behulp van onderstaande aanpassing kan dit commando ook voor de bezitters van bovengenoemde JUNIOR-konfiguratie operationeel gemaakt worden.

De maximale waarden van X en Y (hiervoor mogen uiteraard ook andere variabelenamen gebruikt worden) worden bepaald door het op het moment van gebruik geldende beeldschermformaat.
Bijvoorbeeld:

Bij een beeldschermformaat van 80 posities bij 24 regels zijn de maximale waarden van X en Y respectievelijk 79 en 23.
Overschrijding van deze waarden veroorzaakt een "SYNTAX ERROR".

```

0010 ;*****
0020 ;***
0030 ;*** PRINT&(X,Y) COMMANDO OSI NINE DIGIT BASIC ***
0040 ;***
0050 ;*** JAN VAN HEUVEN ***
0060 ;*** RUNSTRAAT 2 ***
0070 ;*** EINDHOVEN ***
0080 ;***
0090 ;*****
0100 ;
0110 ; VIDEO ROUTINES ELEKTUUR
0120 ;
0130 CLNDN .DE $F0F0 ; verhoog cursor startadres
0140 ADJCUR .DE $F16D ; positioneer cursor
0150 COMCOC .DE $F39C ; reset escape funktie
0160 ;
0170 ; BASIC ROUTINES
0180 ;
0190 CHRGET .DE $00C0 ; haal BASIC character op
0200 CHCKLPAR .DE $0E10 ; test op "("
0210 EVALEXP .DE $1618 ; evalueer expressie
0220 CHCKRPAR .DE $0E0D ; test op ")"
0230 SNERROR .DE $0E1E ; print "SYNTAX ERROR"
0240 BASIC .DE $36C8 ; vervolg PRINT commando
0250 ;
0260 .BA $36A8
0270 ;
0280 NOP
0290 NOP
36AA- 4C B5 25 0300 JMP EXT ; vervolg PRINT&(X,Y) commando

```

36A8- EA

36A9- EA

36AA- 4C B5 25

```

0310 ;
0320 .BA $25B5
0330 ;
25B5- 20 9C F3 0340 EXT JSR COMCOC ; reset escape functie
25B8- 20 C0 00 0350 JSR CHRGET ; haal volgende BASIC char. op
25BB- 20 10 0E 0360 JSR CHCKLPAR ; indien <> "(" dan SN ERROR
25BE- 20 18 16 0370 JSR EVALEXP ; haal waarde op in X-register
25C1- EC D0 EF 0380 CPX $EFD0 ; indien > max. aantal toege-
25C4- B0 3C 0390 BCS ERROR ; stane char. dan SN ERROR
25C6- 8E 00 01 0400 STX $0100 ; bewaar X-register
25C9- C9 2C 0410 CMP #', ; indien volgende basic char.
25CB- D0 35 0420 BNE ERROR ; <> ', ' dan SN ERROR
25CD- 20 C0 00 0430 JSR CHRGET ; haal volgende BASIC char. op
25D0- 20 18 16 0440 JSR EVALEXP ; haal waarde op in X-register
25D3- EC D1 EF 0450 CPX $EFD1 ; indien > max. aantal toege-
25D6- B0 2A 0460 BCS ERROR ; stane reg. dan SN ERROR
25D8- 8E 01 01 0470 STX $0101 ; bewaar X-register
25DB- 20 0D 0E 0480 JSR CHCKRPAR ; volgende char. moet ")" zijn
25DE- AE 00 01 0490 LDX $0100 ; haal X-register weer op
25E1- 8E CB EF 0500 STX $EFCB ; zet dit in positie teller
25E4- AE C0 EF 0510 LDX $EFC0 ; maak cursor start adres
25E7- 8E C2 EF 0520 STX $EFC2 ; gelijk aan VIDEO start adres
25EA- AE C1 EF 0530 LDX $EFC1
25ED- 8E C3 EF 0540 STX $EFC3
25F0- AE 01 01 0550 LDX $0101 ; haal X-register weer op
25F3- 8E CA EF 0560 STX $EFCA ; en zet dit in de regelteller
25F6- 20 F0 F0 0570 REPEAT JSR CLNDN ; verhoog cursor start adres
25F9- CA 0580 DEX ; X-maal het aantal regels
25FA- D0 FA 0590 BNE REPEAT
25FC- 20 6D F1 0600 JSR ADJCUR ; positioneer cursor
25FF- 4C C8 36 0610 JMP BASIC ; en vervolg PRINT commando
0620 ;
2602- 4C 1E 0E 0630 ERROR JMP SNERROR ; "SYNTAX ERROR" exit

```

WAARSCHUWING:

De nieuwe serie CBM-E4 (vanaf medio nov. 1984 heeft van binnen enige wijzigingen ondergaan. Hierdoor kan de bekende OPSTEEKRESETSCHAKELAAR die in de seriele bus wordt gestoken NIET meer worden gebruikt !!!
Doet U dit wel (wat hopelijk nog niet is gebeurd) dan kan bij een reset de computer defekt raken. Het aanbrengen van een resetschakelaar die op de USERSPOORT wordt aangesloten (pen 3 en massa kortstondig aan elkaar) werkt wel goed en U wordt aangeraden voor deze oplossing te kiezen.

PLOTTERPENNEN.

Bent U in het bezit van een plotter b.v. deze op de SHARP MZ700 of de 1520 van COMMODORE dan blijkt dat de plotterpennen moeilijk verkrijgbaar zijn. Radio Shack oftewel TANDY heeft op zijn merk plotters dezelfde pennen in gebruik. De katalogusnummers zijn:

26-1480 voor drie zwarte pennen of
26-1481 voor 1 rode, 1 groene en 1 blauwe

Het belangrijkste is eigenlijk het prijsverschil. Voor een set van 4 pennen van COMMODORE betaalt U Fl. 25,00 De pennen bij TANDY kosten Fl. 8,90 per set van 3 stuks.

D A T B A S

=====

INLEIDING.

=====

DATBAS is een bestands-programma voor de JUNIOR-computer in combinatie met OS65DV3.3 en Elektuur VDU-kaart.

Het programma stelt de gebruiker in staat om zelf bestanden te definieren, records toe te voegen, te wijzigen, te wissen en te printen. Hiertoe wordt gebruik gemaakt van random access disk-files.

PROGRAMMABESCHRIJVING.

=====

De opbouw van het programma is modulair, d.w.z. voor elke functie bestaat een zelfstandige routine hetgeen aanpassing en uitbreiding van het programma sterk vereenvoudigd.

De regels 1-500 omvatten de hoofdroutine van waaruit de diverse opdrachten gegeven kunnen worden.

Na het opstarten van het programma verschijnt het DATBAS-logo en de vraag of men een nieuw bestand wil aanmaken of een bestaand bestand wil inlezen. Indien men voor de eerste mogelijkheid kiest (aanmaken nieuw bestand) moeten vooraf (i.v.m. de beperkingen van OS65D V3.3) datafiles gekreeerd worden, dus buiten het programma om.

Voor elk aan te maken bestand dienen 2 datafiles op de diskette gekreeerd te worden, t.w. :

- 1) Het hoofdbestand waarin de records zullen worden opgeslagen.
- 2) Een hulpbestand waarin DATBAS enige bestandsadministratie bijhoudt, zoals aantal records, aantal velden, lengte der velden.
Voor dit bestand hoeft slechts 1 track gereserveerd te worden.

De naam van het hoofbestand mag men zelf bepalen maar de maximale lengte is beperkt tot 5 tekens.

De naam van hulpbestand 1 heeft dezelfde naam als het hoofdbestand met als toevoeging de letter 'H'.

Nadat men deze files gekreeerd heeft kan men het bestand gaan definieren. Deze bestandsdefinitie is vrij simpel en wordt derhalve hier niet besproken.

Een overzicht van de binnen DATBAS mogelijke opdrachten wordt hieronder gegeven.

BEWAAR

=====

Legt de gevens van een bestand vast op diskette.

Deze opdracht hoeft bij normaal gebruik nooit gegeven te worden omdat als men DATBAS "netjes" beeindigd, alle gegevens op diskette zijn of worden geschreven. Indien men echter een bestand initieel aan het vullen is, is het raadzaam om tussentijds de ingetoetste gegevens te bewaren. Dit voorkomt veel dubbel werk indien het systeem onverhoopt mocht "hangen".

PRINT

=====

Printen van records uit een bestand.

Deze opdracht wordt altijd gevolgd door de vraag "F)ormaat G)een formaat". Voor de uitleg over het formaat printen wordt verwezen naar het betreffende hoofdstuk. Indien men voor "G)een formaat" kiest, volgt de vraag "T)otaal overzicht R)ecord overzicht".

T)otaal overzicht houdt in dat men zelf aangeeft welke velden (en in welke volgorde) van de geselecteerde records afgedrukt zullen worden.

Deze velden worden dan in 1 printregel afgedrukt.

Per nieuwe pagina worden de regels voorafgegaan door 3 kopregels.

R)ecord overzicht houdt in dat alle velden van de geselecteerde records afgedrukt worden volgens een default formaat.

Vervolgens volgt de vraag "S)cherm P)rinter".

Indien men voor "P)rinter" kiest moeten de regellengte en het aantal regels per pagina opgegeven worden.

De printroutine loopt vervolgens via de selectie-routine waarbij men de mogelijkheid heeft om maximaal 8 selectiekriteria op te geven waaraan de te printen records moeten voldoen.
De selectiemogelijkheden zijn:

- 1) Recordnummer
- 2) Veldnummer
- 3) Begin

Nadat de selectiekriteria zijn opgegeven (of niet natuurlijk) kan men het printen starten met "Begin".

N.B.
=====

M.b.t. het opgeven van de selectiekriteria is het volgende nog van belang. Stel men heeft een bestand gedefinieerd (en gevuld) waarin het veld NAAM voorkomt. Indien men nu alle namen wil selecteren die met de letter "H" beginnen, kan men opgeven "H*". Ook is het mogelijk om meerdere letters op te geven, bijvoorbeeld "HEU*" zal alle namen selecteren die met "HEU" beginnen.

TOEVOEGEN

=====

Voegt records toe aan een bestand.
De in te vullen velden worden op het scherm weergegeven en kunnen vervolgens ingevuld worden.

VERANDER

=====

Veranderen van record(s) in het bestand.
Ook deze opdracht loopt via de selectie-routine (zie PRINT-opdracht). Nadat het te wijzigen record is gevonden, bevindt de cursor zich achter het eerste veld van het record.
Nu kan men kiezen voor:

- 1 = aanpassen inhoud veld.
- 2 = niet aanpassen inhoud veld, dus door naar het volgende veld.
- 3 = naar einde record.

WISSEN

=====

Wissen van record(s) in het bestand.
Ook deze opdracht loopt weer via de selectie-routine (zie PRINT-opdracht). Nadat het te wissen record is gevonden heeft men de mogelijkheid om het record te laten vervallen (of niet natuurlijk).

N.B.
=====

De gewiste records laten uiteraard lege plaatsen achter op de diskette. Deze lege plaatsen worden echter "geregistreerd" door het programma, zodat toe te voegen records in eerste instantie op plaatsen van gewiste records terecht komen. Dit leidt tot een zo efficiënt mogelijk gebruik van de beschikbare disk-ruimte.

SORTEER

=====

Sorteert records in het bestand.
Sortering vindt plaats op basis van op te geven veld, en kan in toenemende of afnemende volgorde plaatsvinden.

NIEUW

=====

Sluit het gebruikte bestand en geeft de gebruiker de mogelijkheid om met een ander bestand verder te gaan of om een nieuw bestand te definiëren.

EINDE

=====

Sluit het gebruikte bestand en beëindigt DATBAS.

FORMAAT PRINTEN.

Indien men na de print-opdracht kiest voor "F)ormaat" vraagt het programma of een formaat aangemaakt dient te worden of dat een op diskette opgeslagen formaat gebruikt dient te worden.

In het laatste geval zullen de te printen records dus in dat formaat geprint worden.

Het aanmaken van een formaat verloopt als volgt:

Men kan kiezen uit:

- 1) Veldnaam : de naam van een veld zal geprint worden (de titel als het ware).
- 2) Veldinhoud : de inhoud van een veld zal geprint worden.
- 3) Spaties : een op te geven aantal spaties zal geprint worden.
- 4) Nieuwe regel : een op te geven aantal regels zal geprint worden.
- 5) Tekst : op te geven tekst zal geprint worden.
- 6) Einde : beeindigen formaatdefinitie, formaat wordt weggeschreven naar diskette.

Voorbeeld van het gebruik:

Bestand met 3 velden: NAAM, ADRES en WOONPLAATS.
Gewenst formaat (">" zijn blanko regels):

```
>
>
> Naam
> Adres
>
> Woonplaats
>
>
> Tekst.
```

Men definieert dit formaat als volgt:

| Keuze: | Toets in: | Verklaring: |
|--------------|-----------|-----------------------------|
| V,I,S,N,T,E | N | Nieuwe regel |
| Hoeveel ? | 3 | 2 blanko regels |
| V,I,S,N,T,E | S | Spaties |
| Hoeveel ? | 3 | 3 spaties |
| V,I,S,N,T,E | I | inhoud van een veld |
| Veldnummer ? | 1 | veldnr. van veld NAAM |
| V,I,S,N,T,E | N | Nieuwe regel |
| Hoeveel ? | 1 | vervolg op nieuwe regel |
| V,I,S,N,T,E | S | Spaties |
| Hoeveel ? | 3 | 3 spaties |
| V,I,S,N,T,E | I | inhoud van een veld |
| Veldnummer ? | 2 | veldnr. van veld ADRES |
| V,I,S,N,T,E | N | Nieuwe regel |
| Hoeveel ? | 2 | 1 blanko regel |
| V,I,S,N,T,E | S | Spaties |
| Hoeveel ? | 3 | 3 spaties |
| V,I,S,N,T,E | I | inhoud van een veld |
| Veldnummer ? | 3 | veldnr. van veld WOONPLAATS |
| V,I,S,N,T,E | N | Nieuwe regel |
| Hoeveel ? | 3 | 2 blanko regels |
| V,I,S,N,T,E | S | Spaties |
| Hoeveel ? | 3 | 3 spaties |
| V,I,S,N,T,E | T | Tekst opgeven |

Nu kan de tekst opgegeven worden.

Na een carriage return keert men terug naar de keuzemogelijkheden.

Beeindigen van de formaatdefinitie gaat door middel van keuze "E".

Vervolgens kan men het aangemaakte formaat op diskette bewaren onder een vrij te kiezen naam. Wel dient men er uiteraard voor te zorgen dat deze bestandsnaam reeds vooraf gekreeert en geinitialiseert is.

```

1 REM*** JUNIOR DATABASE MANAGEMENT SYSTEEM *****
2 REM*** DOOR J.V.HEUVEN. EINDHOVEN AUGUSTUS 1984*****
3 POKE2888,0:POKE8722,0:POKE2797,32:GOTO100
5 PRINT!(49)::RETURN
7 PRINT&(0,4)!(52)::RETURN
9 PRINT!(51)::RETURN
10 IFRL%<9THENP1=3:P2=255:GOTO16
11 IFRL%<17THENP1=4:P2=128:GOTO16
12 IFRL%<33THENP1=5:P2=64:GOTO16
13 IFRL%<65THENP1=6:P2=32:GOTO16
14 IFRL%<129THENP1=7:P2=16:GOTO16
15 P1=8:P2=8
16 RETURN
20 PRINT&(X,0)!(56)::INPUTA$:A$=LEFT$(A$,1):SR=1
22 IFA$=MID$(D$,SR,1)THEN28
24 SR=SR+1:IFSR>LEN(D$)THEN20
26 GOTO22
28 RETURN
30 PRINT&(0,0)"Let op!!! het gewenste overzicht past niet op ";
32 PRINT"de printregel. Toets return.":INPUT D$:PRINT&(0,5)::RETURN
34 FOR J=0 TO NH
36 PRINT&(20,J+5)"(:J+1:)":H$(J)
38 NEXT:RETURN
100 DISK!"60 f330":DISK!"CALL DOAO=08,1"
105 DIMC$(7),C1$(7),C2$(7),F$(5):CH=0
110 PRINT&(0,0)"Kies: A)anmaken bestand I)nlezen bestand S)toppen "
120 X=55:D$="AIS":GOSUB20
130 IFSR=3THENEND
140 ONSRGOSUB1500,1000
200 GOSUB7:PRINT&(0,0)!(56)"Kies: B)ewaar P)rint T)oevoegen ";
210 PRINT"V)erander W)is S)orteer N)ieuw E)inde"
220 X=75:D$="BPTVWSNE":GOSUB20
320 IF SR>6THEN500
330 ONSRGOSUB2000,3000,4000,350,360,7000:GOTO200
350 SB=3:GOTO8000
360 SB=4:FS=1:GOTO8000
500 IFSS=1THEN530
520 GOSUB2000
530 DISK CLOSE,6
540 IFSR=7THENCLEAR:GOSUB5:GOTO105
550 END
1000 PRINT&(0,0)!(56)"Naam van het bestand ? ":INPUTF$
1005 F$=LEFT$(F$,5):A$=F$
1010 TRAP1230:DISK OPEN,6,F$+"H":INPUT#6,NH,NI,MX,LK,RL%
1020 GOSUB10
1130 DIMH$(NH),L$(NH),I$(MX),P$(MX),TI$(NH),KR((NH+1)*10)
1140 FORI=0TONH:INPUT#6,H$(I),L$(I):NEXT
1160 IFNI=-1THEN1180
1170 FORI=0TONI:INPUT#6,P$(I):NEXT
1180 DISK CLOSE,6
1185 IFNI=-1THEN1220
1190 DISK OPEN,6,F$:POKE12076,P1:POKE12042,P2
1210 GOSUB1300
1215 GOSUB5:PRINT&(0,1)::FORX=1TO80:PRINTCHR$(129)::NEXT:PRINT
1217 SS=1:TRAP0:GOSUB1220:GOTO1225
1220 PRINT&(0,2)"Bestandsnaam :":F$;",";NI+1;" records"
1223 PRINT&(67,2):NH+1;" Velden":RETURN
1225 PRINT&(0,3)::FORX=1TO80:PRINTCHR$(135)::NEXT:PRINT:RETURN

```

```

1230 PRINT$(0,1)!(56):PRINT$(0,0)!(56)"Bestand ":A$;
1236 PRINT" niet gevonden ! Toets <RETURN> ";;INPUTA$:TRAP0:GOTO110
1300 FORI=0TONI:DISK GET,P$(I):FORX=0TOCH:INPUT#6,I$(I):NEXT:NEXT
1340 DISK CLOSE,6:RETURN
1500 PRINT$(0,0)!(56)"Naam van het nieuwe bestand ? "
1507 PRINT$(33,0)::INPUTF$:IFF$=""THEN1507
1510 F$=LEFT$(F$,5)
1512 TRAP1230:A$=F$:GOSUB1662:A$=F$+"H":GOSUB1662:TRAP0
1525 PRINT$(0,0)!(56)"Hoeveel velden ? "
1527 PRINT$(20,0)::INPUTA$:NH=VAL(A$):IFNH<1THEN1527
1540 NH=NH-1:NI=-1:LK=-1
1545 DIMH$(NH),L$(NH),TI$(NH)
1550 GOSUB5:PRINT$(0,1)::FORX=1TO80:PRINTCHR$(129)::NEXT:PRINT
1560 PRINT$(0,2)"Bestandnaam : ";F$::PRINT$(69,2);NH+1;" Velden"
1564 PRINT$(0,3)::FORX=1TO80:PRINTCHR$(135)::NEXT:PRINT
1566 PRINT$(0,4)"VELDNR. ";:PRINT$(9,4)"VELDNAAM";:PRINT$(60,4)"LENGTE"
1570 RT=0:FORI=0TONH:IFRT>15THENRT=0:PRINT$(0,5)!(52)
1580 RT=RT+1:PRINT$(3,RT+5)I+1:PRINT$(7,RT+5)::INPUTH$(I)
1585 PRINT$(61,RT+5)::INPUTA$:L$(I)=INT(VAL(A$)):IFL$(I)<1THEN1585
1595 RL%=RL%+L$(I)+1:NEXT:GOSUB10
1620 PRINT$(0,0)!(56)"Welk veldnummer is het langste ";
1623 PRINT"waarop je wil sorteren ? "
1625 PRINT$(57,0)!(56)::INPUTA$:J=VAL(A$):J=J-1:IFJ<0ORJ>NHTHEN1625
1630 J=L$(J):MX=INT((FRE(0)-2000)/(J+2))
1640 DIMI$(MX),P$(MX)
1660 SS=0:RETURN
1662 DISK OPEN,6,A$:DISK CLOSE,6:RETURN
1700 R=P$(I)
1710 IFREPEAT=1THEN1715
1712 DISK OPEN,6,F$:POKE12076,P1:POKE12042,P2:REPEAT=1
1715 DISK GET,R
1720 FORJ1=0TONH
1730 INPUT#6,TI$(J1)
1740 NEXT:TRAP0
1750 RETURN
1800 IFREPEAT=0THENTRAP2290:DISK OPEN,6,F$
1802 POKE12076,P1:POKE12042,P2
1805 DISK GET,R
1810 FORJ1=0TONH
1820 PRINT#6,TI$(J1)
1830 NEXT:TRAP0
1840 RETURN
2000 TRAP2290
2010 DISK OPEN,6,F$+"H"
2020 PRINT#6,NH
2030 PRINT#6,NI:PRINT#6,MX:PRINT#6,LK:PRINT#6,RL%
2040 FORI=0TONH:PRINT#6,H$(I):PRINT#6,L$(I):NEXT
2060 IFNI=-1THEN2270
2070 FORI=0TONI
2080 PRINT#6,P$(I)
2090 NEXT
2270 DISK CLOSE,6:TRAP0
2280 SS=1:RETURN
2290 GOSUB7:PRINT$(0,0)!(56)"D I S K   E R R O R   TOETS <RETURN> ";
2295 INPUTA$:TRAP0:SS=1:GOTO200
3000 GOSUB7
3003 IFNI=-1THENGOSUB9000:RETURN
3005 PRINT$(0,0)!(56)"Kies: F)ormaat G)een formaat "
3006 X=32:D$="FG":GOSUB20

```

```

3010 FS=1:IFSR=1THENGOSUB10000:FS=2
3012 SB=1:IFFS=2THENTL=0:GOTO3040
3014 PRINT$(0,0)!(56)"Kies: T)otaal overzicht R)ecord overzicht "
3015 X=48:D$="TR":GOSUB20
3019 TL=0:IFSR=2THEN3040
3020 TL=1:GOSUB 34
3022 PRINT$(0,0)!(56)"Toets veldvolgorde in d.m.v. veldnummer=":
3023 PRINT"gescheiden door / "
3024 PRINT$(60,0)!(56)::INPUTA$:Q=0
3026 FORJ=1TOLEN(A$)
3028 IFMID$(A$,J,1)="/"THEN3036
3030 X=VAL(MID$(A$,J,1))
3031 IFVAL(MID$(A$,J,2))<>0THENX=VAL(MID$(A$,J,2)):J=J+1
3032 FO=0:IFX<10RX>NH+1THENFO=1:J=LEN(A$):GOTO3036
3034 KR(Q)=X-1:Q=Q+1
3036 NEXT:IFFO=1THEN3024
3038 Q=Q-1:GOSUB7
3040 PRINT$(0,0)!(56)"Kies: S)chem P)rinter "
3042 X=27:D$="SP":GOSUB20:AR=15:RL=79
3043 IFSR=1THENDV=1:GOTO3054
3044 DV=4:PRINT$(0,0)!(56)"Regellengte ? "
3045 PRINT$(15,0)!(56)::INPUTA$:RL=VAL(A$)
3047 IFRL<10RRL>150THEN3045
3048 PRINT$(0,0)!(56)"Aantal regels per pagina ?"
3050 PRINT$(28,0)!(56)::INPUTA$:AR=VAL(A$)
3052 IFAR<10RAR>100THEN3050
3054 GOSUB8010:RETURN
3300 ONF6GOSUB3700,3800
3303 IFSB=40RTL=10RDV<>1THEN3320
3305 PRINT$(0,0)!(56)"Toets <RETURN> voor vervolg of ":
3306 PRINT"<S> voor stoppen ":INPUTA$
3310 IFA$="S"THENRS=1
3320 RETURN
3700 IFTL=1THEN3715
3702 IFDV=1THENGOSUB7:PRINT$(0,5);
3703 PRINT#DV,"RECORD <";I+1;">"
3704 FOR J=0TONH
3705 IFDV=1THENPRINT$(0,J+8);
3706 PRINT#DV,H$(J)::PRINT#DV,TAB(15)" : ";TI$(J)
3708 NEXT
3710 RETURN
3715 IFRT<ARTHEN3728
3716 IFDV=1THENGOSUB7
3717 RT=3:RR=0:KQ$="":FORJ=0TOQ
3718 RR=RR+L%(KR(J))
3719 NEXT:RR=INT((RL-RR)/(J)):IFRR<=0THENRR=1
3720 FORJ=0TOQ:A$=H$(KR(J))
3721 IFLEN(A$)>=L%(KR(J))+RRTHENA$=LEFT$(A$,L%(KR(J))+RR-1)+" "
3722 IFLEN(A$)<L%(KR(J))+RRTHENA$=A$+" ":GOTO3722
3723 A$=LEFT$(A$,L%(KR(J))+RR):KQ$=KQ$+A$
3724 NEXT:IFDV=1THENPRINT$(0,4);
3725 FORX=1TORL:PRINT#DV,"="::NEXT:PRINT#DV:IFLEN(KQ$)>RLTHENGOSUB30
3726 KQ$=LEFT$(KQ$,RL):PRINT#DV,KQ$
3727 FORX=1TORL:PRINT#DV,"="::NEXT:PRINT#DV
3728 R$="":FORJ=0TOQ:IFKR(J)=-1THENJ=NH:GOTO3731
3729 IFTI$(KR(J))=""THENFORX=1TOL%(KR(J)):R$=R$+" ":NEXT
3730 R$=R$+TI$(KR(J)):FORX=1TORR:R$=R$+" ":NEXT
3731 NEXT

```

```

3732 R#=LEFT$(R$,RL)
3734 PRINT#DV,R$:RT=RT+1:IFRT>=ARANDDV=1THENGOSUB3305
3736 RETURN
3800 IFDV=1THENGOSUB7
3802 J=1:T=0:B$="":RT=1
3820 J1=VAL(MID$(F$(T),J,1)):J=J+1
3830 IFJ1<5THENN=VAL(MID$(F$(T),J,2)):J=J+2
3840 ONJ1GOTO3850,3860,3870,3890,3910,3970
3850 A#=H$(N):GOTO3950
3860 A#=TI$(N):GOTO3950
3870 IFLEN(B$)+N>70THEN3960
3871 FORJ2=1TON
3879 B#=B$+" ":NEXT
3880 GOTO3960
3890 PRINT#DV,B$:RT=RT+1
3895 IFN>1THENFORJ2=2TON:PRINT:NEXT
3900 B$="":GOTO3960
3910 IFJ>LEN(F$(T))THENT=T+1:J=1
3920 J2=J
3930 IFMID$(F$(T),J2,1)<>"!"THENJ2=J2+1:GOTO3930
3940 A#=MID$(F$(T),J,J2-J):J=J2+1
3950 B#=B$+A#
3960 IFJ>LEN(F$(T))THENT=T+1:J=1
3965 GOTO3820
3970 PRINT#DV,B$:RT=RT+1
3972 IFDV<>1THENFORX=RTTOAR:PRINT#DV:NEXT
3974 RETURN
4000 GOSUB7:SS=0:NI=NI+1
4005 PRINT:PRINT"RECORD <";NI+1;">"
4010 FORJ=0TONH:PRINT&(1,J+7):H$(J)::PRINT&(26,J+7)": ";
4012 T=L%(J)
4014 A$="":FORX=1TOT:A#=A#+CHR$(127):NEXT:PRINTA$:NEXT:PRINT
4018 FORJ=0TONH:PRINT&(28,J+7)::INPUTTI$(J):GOSUB4520:NEXT
4040 IFLK=-1THENR=NI:GOTO4080
4050 R=LK
4060 DISK OPEN,6,F$:POKE12076,P1:POKE12042,P2
4065 DISK GET,R:INPUT#6,LK
4070 DISK CLOSE,6
4080 GOSUB1800:GOSUB1220:P%(NI)=R:I$(NI)=TI$(CH)
4085 PRINT&(0,0)!(56)"Wilt u nog meer records toevoegen (J/N) ? ";
4087 PRINT&(41,0)!(56)::INPUTA$:IFA$<>"J"ANDA$<>"N"THEN4087
4088 IFA$="J"THENREPEAT=1:GOTO4000
4089 DISK CLOSE,6:REPEAT=0
4090 RETURN
4500 PRINTH$(J)::PRINTTAB(25)": ":INPUTTI$(J)
4520 T=L%(J)
4540 IFLEN(TI$(J))>TTHENTI$(J)=LEFT$(TI$(J),T)
4542 IFLEN(TI$(J))<TTHENTI$(J)=TI$(J)+" ":GOTO4542
4550 RETURN
5000 GOSUB7
5005 PRINT&(0,0)!(56)"1 = Aanpassen, 2 = Niet aanpassen, ";
5006 PRINT"3 = Naar einde record"
5030 PRINT&(0,5)"RECORD <";I+1;">"
5040 CS=1:RS=0:FORJ=0TONH
5050 PRINT&(0,J+7):H$(J)::PRINT&(25,J+7)": ":TI$(J):NEXT
5052 FORJ=0TONH
5053 PRINT&(75,J+7)!(56)::INPUTA$:IFA$<"1"ORA$>"3"THEN5053
5055 IFA$="2"THEN5090
5061 IFA$="3"THENJ=NH:GOTO5090

```

```

5079 PRINT$(0,J+7)::PRINT$(56);
5080 GOSUB 4500
5085 CS=0
5090 NEXT J
5095 RS=0
5100 IFCS=0THENGOSUB1800:I$(I)=TI$(CH)
5110 RETURN
6000 PRINT$(0,0)!(56)"Moet dit record vervallen (J/N) "
6070 PRINT$(32,0)!(56)::INPUTA$:IFA$<>"J"AND A$<>"N"THEN6070
6080 IFA$="N"THEN6150
6095 DISK GET,P$(I)
6097 PRINT#6,LK
6110 LK=P$(I)
6120 FORI1=I+1TONI
6130 I$(I1-1)=I$(I1):P$(I1-1)=P$(I1)
6135 NEXT
6140 NI=NI-1:SS=0:I=I-1:GOSUB1220
6150 RETURN
7000 IFNI=-1THENGOSUB9000:RETURN
7005 GOSUB7:GOSUB34
7040 PRINT$(0,0)!(56)"Op welk veldnummer wil je sorteren "
7041 PRINT$(34,0)!(56)::INPUTA$:J1=VAL(A$):J1=J1-1
7050 IFJ1<0ORJ1>NHTHEN7041
7055 IFJ1<>CHTHENCH=J1:GOSUB1300
7060 PRINT$(0,0)!(56)"Kies: T)oenemend A)fnemend "
7061 X=28:D$="TA":GOSUB20
7080 GOSUB7:PRINT$(0,0)!(56)"*** EVEN GEDULD A.U.B. ***"
7100 FORI=0TONI-1
7110 T=I
7120 FORI1=T+1TONI
7125 DNSRGOTO7130,7140
7130 IFI$(I1)<I$(T)THENT=I1
7135 GOTO7145
7140 IFI$(I1)>I$(T)THENT=I1
7145 NEXT
7150 IFT=ITHEN7180
7160 T$=I$(T):I$(T)=I$(I):I$(I)=T$
7170 J1=P$(T):P$(T)=P$(I):P$(I)=J1
7180 NEXT
7200 SS=0:RETURN
8000 IFNI=-1THENGOSUB9000:RETURN
8010 I1=0:I2=NI:J=0:C1%(0)=-1:BS=1:REPEAT=0
8015 GOSUB7:PRINT$(0,0)!(56)"Selektiemogelijkheden: ";
8016 PRINT"R)ecordnummer V)eldnummer B)egin "
8020 X=59:D$="RVB":GOSUB20
8028 I=SR-1:IFSR<>2THEN8045
8030 FORI=0TONH:PRINT$(10,I+5)" T. ";I+1;"> ";H$(I):NEXT
8035 PRINT$(0,0)!(56)"Welk veldnummer ? "
8037 PRINT$(18,0)!(56)::INPUTA$:I=VAL(A$)
8042 IFI<1ORI>NH+1THEN8037
8045 IFSR=3THENC1%(J)=-1:GOTO8150
8050 C1%(J)=I-1
8054 IFSR=1THENPRINT$(0,0)!(56)"Moet recordnummer ":GOTO8060
8056 PRINT$(0,0)!(56)"Moet het veld ";H$(I-1);" ";
8060 PRINT"<, = of > zijn dan de in te geven waarde ? "
8061 PRINT$(75,0)!(56)::INPUTA$:A$=LEFT$(A$,1):SR=1
8063 IF A$=MID$("<=>".SR,1)THEN8070

```

```

8065 SR=SR+1:IFSR>3THEN8061
8067 GOTO8063
8070 C2%(J)=SR
8080 PRINT&(0,0)!(56)"Vergelijken met: ";
8085 IFC1%(J)=-1THEN8100
8090 INPUT" ";C$(J):J=J+1
8092 IFJ>7THEN8160
8095 GOTO8015
8100 PRINT&(18,0)!(56);:INPUT" ";A$:I=VAL(A$)
8102 IFI<10RI>NI+1THEN8100
8105 I=I-1
8110 IFC2%(J)=1THENI2=I
8120 IFC2%(J)=2THENI1=I:I2=I
8130 IFC2%(J)=3THENI1=I
8135 J=J+1:IFJ>7THEN8160
8140 GOTO 8015
8150 IFJ<2THEN8200
8160 PRINT&(0,0)!(56)"Kies: E)n relatie  O)f relatie ";
8161 X=32:D$="E0":GOSUB20:BS=SR
8200 RS=0:J1=C1%(0)
8210 DS=0:FOR J=0TO7
8220 IFC1%(J)=-1THENJ=7:GOTO8240
8230 IFJ1<>C1%(J)THENJ1=-2
8240 NEXT
8245 IFJ1>-1ANDJ1<>CHTHENCH=J1:GOSUB1300
8246 IFJ1=-2THENDS=1
8250 I=I1-1:RT=100:FORI3=I1TOI2:I=I+1
8251 IFDS=0THENTI$(CH)=I$(I):GOTO8255
8252 GOSUB1700
8255 AS=0:FORJ=0TO7
8260 IFC1%(J)=-1THENJ=7:GOTO8345
8270 ONC2%(J)GOTO8280,8290,8310
8280 IFTI$(C1%(J))<=C$(J)THEN8330
8285 GOTO8340
8290 IFLEFT$(TI$(C1%(J)),LEN(C$(J)))=C$(J)THEN8330
8295 IFRIGHT$(C$(J),1)<>"*"THEN8340
8298 T=LEN(C$(J))-1
8300 IFLEN(TI$(C1%(J)))<TTHEN8340
8302 IFLEFT$(TI$(C1%(J)),T)=LEFT$(C$(J),T)THEN8330
8305 GOTO8340
8310 IFTI$(C1%(J))>=C$(J)THEN8330
8320 GOTO8340
8330 IFBS=2THENAS=1:J=7
8335 GOTO8345
8340 IFBS=1THENAS=2:J=7
8345 NEXTJ
8350 IFAS=0ANDBS=1THEN8355
8352 IFAS<>1THEN 8380
8355 IFDS=0THENGOSUB1700
8360 IFBS<>3THENGOSUB3300
8365 IFBS=3THENGOSUB5000
8370 IFBS=4THENGOSUB6000
8375 IFRS=1THENI3=I2
8380 NEXTI3
8385 DISK CLOSE,6:REPEAT=0:TL=0
8390 PRINT&(0,0)!(56)"Dat was alles, toets <RETURN> ";;INPUTA$
8395 RETURN
9000 PRINT&(0,0)!(56)"Er zijn geen records aanwezig !!!! Toets ";
9010 PRINT"<RETURN> ";;INPUTA$:RETURN

```

```

10000 GOSUB7: IFF$(0)="" THEN 10042
10010 PRINT$(0,0)!(56)"Wil je hetzelfde formaat gebruiken zoals ";
10020 PRINT"je hiervoor gebruikt hebt ? (J/N)";
10022 X=77:D$="JN":GOSUB20
10025 IFSR=1 THEN RETURN
10042 PRINT$(0,0)!(56)"Kies: I)nlezen formaat A)anmaken formaat ";
10044 X=42:D$="IA":GOSUB20
10050 IFSR=2 THEN 10200
10052 PRINT$(0,0)!(56)"Welke formaatnaam ? ";; INPUTA$
10090 TRAP10170:PRINT$(0,4);
10100 DISK OPEN,6,A$:INPUT#6,NF
10140 FORJ=0 TO NF:INPUT#6,F$(J):NEXT
10150 DISK CLOSE,6:TRAP0:RETURN
10170 PRINT$(0,0)!(56)"Formaat niet gevonden !! toets <RETURN>";
10180 INPUTA$:GOTO200
10200 NF=0:J=0:F$(0)=""
10205 PRINT$(0,0)!(56)"Kies: V)eldnaam I)inhoud S)paties N)ieu";
10210 PRINT"we regel T)ekst E)inde ";
10215 X=67:D$="VISNTE":GOSUB20:A$=RIGHT$(STR$(SR),1)
10240 F$(NF)=F$(NF)+A$:J=J+1:J1=SR
10250 ONJ1GOTO10260,10260,10300,10300,10350,10400
10260 FORT=0 TO NH:PRINT$(20,6+T)!(56)T+1;" ";H$(T):NEXT
10270 PRINT$(0,0)!(56)"Welke ? ";
10272 PRINT$(8,0)!(56)";: INPUTA$:T=VAL(A$)
10275 T=T-1:IFT<0ORT>NH THEN 10272
10280 B$=STR$(T):IFLEN(B$)=1 THEN B$="0"+B$
10285 GOTO10320
10300 PRINT$(0,0)!(56)"Hoeveel ? ";
10302 PRINT$(9,0)!(56)";: INPUTB$
10304 T=VAL(B$):IFLEN(B$)=1 THEN B$="0"+B$
10305 IFT<1ORT>99 THEN 10302
10320 F$(NF)=F$(NF)+B$:J=J+1
10330 GOTO10380
10350 GOSUB7:PRINT$(0,6)"TEKST :";: INPUTB$:B$=B$+"!"
10360 IFLEN(B$)+J>70 THEN NF=NF+1:J=0:F$(NF)=""
10370 F$(NF)=F$(NF)+B$:J=J+LEN(B$)
10380 IFJ>67 THEN NF=NF+1:J=0:F$(NF)=""
10390 GOSUB7:GOTO10205
10400 PRINT$(0,0)!(56)"Onder welke naam moet het formaat bewaart ";
10402 PRINT"worden ?"
10404 PRINT$(52,0)!(56)";: INPUTA$:A$=LEFT$(A$,6)
10418 TRAP10500:PRINT$(0,4);:DISK OPEN,6,A$
10420 PRINT#6,NF:FORJ=0 TO NF:PRINT#6,F$(J):NEXT
10440 DISK CLOSE,6:TRAP0:RETURN
10500 PRINT$(0,0)!(56)"Bestand ";A$;" bestaat niet ! ";
10510 PRINT"Toets <RETURN> ";; INPUTA$:GOTO200

```

Ik zoek alle technische specificaties over de MIDI-bus, en schema's over reeds bestaande interfaces voor koppeling van MIDI-synthesizers, zoals de DX7 van YAMAHA of de JX3P, JX8P, Jupiter 6 en Juno 106 van Roland aan computers. Wie kan mij daaraan helpen? Stefan Sierling, Wavensesteenweg 88 A, B-1990 Hoeillaart.

TE KOOP AANGEBODEN

| | |
|--|--------------------|
| 8" FDOS controllerkaart met FD 1771 | fl. 90.00 |
| 2 x 8K Ram/Rom kaart met 2 x 8K Ram en in Eorom's FDOS 1.2 voor 8" | fl. 100.00 |
| Verkoop ten behoeve van een Belgisch lid. | |
| Voor informatie of bezichtiging: Tel.: 01807 - 19881 | |
| Verzendkosten zijn niet in de prijs inbegrepen. | |
| ITT modem 600/1200 baud, half duplex | t.e.a.b. |
| 5" Shugart SA 200 floppy-drive, 40 tr. s.s. | t.e.a.b. |
| IBM 029 kaartoonsmachine | t.e.a.b. |
| Olympia schrijfmachine met brede wagen, niet-electrisch, met lilliput-letter | t.e.a.b. |
| A. Mueller | |
| Sing. Semeynstraat 78 1-H | |
| 1061 GH Amsterdam | Tel.: 020 - 860245 |

ASS L

```

0005      .LS
0010      * * * * *
0020      *
0030      * CENTRONICS PRINTER INTERFACE *
0040      *
0050      * DEVICE 4 OR 5 ON COMMODORE 64 *
0060      *
0070      * BY: RUUD UPHOFF FEB. 1984 *
0080      *
0090      * * * * *
0100
0110 ENTRY .BA $C000      :RESIDING AT $C000
0120      .OS
0130
0140 !!!REM .MD :REMARK MNEMONIC
0150      .ME
0160
0170      ***** EXTERNAL PROCEDURES *****
0180
0190      :PROCEDURE "OUTP"
0200      : -PRINT CHARACTER ON CURRENT OUTPUT DEVICE
0210      : -ENTERED BEYOND VECTOR
0220 OUTP   .DE $F1CA      :ADDRESS AT $F1CA
0230
0240      :PROCEDURE "DESELECT"
0250      : -DESELECT ALL I/O DEVICES AND SET DEFAULT I/O
0260      : -INPUT FROM KEYBOARD AND OUTPUT TO SCREEN
0270 DESELECT .DE $F333      :ADDRESS AT $F333
0280
0290      :PROCEDURE "SELECT"
0300      : -SELECT OUTPUT DEVICE FROM FILE STORED IN "DEVI"
0310      : -CMD DEVICE WITH SECONDARY ADDRESS AS STORED
0320      : IN "SECA"
0330      : -THEN ALL OUTPUT TO THAT DEVICE
0340 SELECT .DE $F250      :ADDRESS AT $F250
0350
0360      ***** EXTERNAL VARIABLES *****
0370
0380      :I/O REGISTERS OF 6526 CIA #1
0390
0400 DRA     .DE $DD00      :DATA REGISTER A
0410 DRB     .DE $DD01      :DATA REGISTER B
0420 DDRA    .DE $DD02      :DATA DIRECTION REGISTER A
0430 DDRB    .DE $DD03      :DATA DIRECTION REGISTER B
0440
0450      :ALIAS FOR SOME OF ABOVE'S REG'S
0460
0470 BUSY     .DE DRA        :ALIAS OF DRA
0480 BUS      .DE DRB        :ALIAS OF DRB
0490
0500      :KERNAL VECTORS TO ABOVE PROCEDURES
0510
0520 OUTV     .DE $0326      :VECTOR TO "OUTP"
0530 DESELECTV .DE $0322      :VECTOR TO "DESELECT"
0540 SELECTV  .DE $0320      :VECTOR TO "SELECT"
0550
0560
0570      :KERNAL PAGE ZERO
0580
0590 DEVI     .DE $00BA      :FILE DEVICE NUMBER
0600 SECA     .DE $00B9      :FILE SEC. ADDRESS
0610
0620      ***** INITIALIZE BY SYS 49152 *****
0630
0640 INIT     REM           :PROC "INIT"
0650          LDA #L,SCAN    : SET LOW BYTE OF SELECT VECTOR
0660          STA SELECTV     : TO AN INTERCEPT ROUTINE
0670          LDA #H,SCAN    : SAME ACTION
0680          STA SELECTV+1   : FOR HIGH BYTE
0690          LDA #L,RESET    : SET LOW BYTE OF DESELECT VECTOR

```

C000- A9 40
C002- 8D 20 03
C005- A9 C0
C007- 8D 21 03
C00A- A9 2D

| | | | | | | | |
|-------|----|----|----|------|-----------------|---|--|
| CO0C- | 8D | 22 | 03 | 0700 | STA DESELECTV | : | TO AN INTERCEPT ROUTINE |
| CO0F- | A9 | C0 | | 0710 | LDA #H, RESET | : | SAME ACTION |
| CO11- | 8D | 23 | 03 | 0720 | STA DESELECTV+1 | : | FOR HIGH BYTE |
| CO14- | 60 | | | 0730 | RTS | : | ENDPROC |
| | | | | 0740 | | : | |
| | | | | 0750 | | : | |
| | | | | 0760 | | : | ***** KERNAL EXPANSION ***** |
| | | | | 0770 | | : | |
| | | | | 0780 | | : | PROCEDURE "SETCENTR" |
| | | | | 0790 | | : | -DESELECT ALL CURRENT I/O DEVICES |
| | | | | 0800 | | : | -SET ALL OUTPUT TO CENTRONICS BUS |
| | | | | 0810 | | : | -INITIALIZE CENTRONICS BUS |
| | | | | 0820 | SETCENTR | : | |
| CO15- | A9 | 6B | | 0830 | REM | : | PROC "SETCENTR" |
| CO17- | 8D | 26 | 03 | 0840 | LDA #L, PROUT | : | SET HIGH BYTE FROM OUTPUT VECTOR |
| CO1A- | A9 | C1 | | 0850 | STA OUTV | : | TO CENTRONICS ROUTINE |
| CO1C- | 8D | 27 | 03 | 0860 | LDA #H, PROUT | : | SAME ACTION |
| CO1F- | A9 | FF | | 0870 | STA OUTV+1 | : | FOR HIGH BYTE |
| CO21- | 8D | 03 | DD | 0880 | LDA #\$FF | : | 8 BITS FOR DATA OUTPUT |
| CO24- | AD | 02 | DD | 0890 | STA DDRB | : | INTO DDRB |
| CO27- | 29 | FB | | 0900 | LDA DDRA | : | GET STATE OF DDRA |
| CO29- | 8D | 02 | DD | 0910 | AND #\$FB | : | SET BIT 2 FOR OUTPUT |
| CO2C- | 60 | | | 0920 | STA DDRA | : | AND RESTORE INTO DDRA |
| | | | | 0930 | RTS | : | ENDPROC |
| | | | | 0940 | | : | |
| | | | | 0950 | | : | PROCEDURE "RESET" (CALLED BY KERNAL NOW) |
| | | | | 0960 | | : | -RESTORE ORIGINAL OUTPUT VECTOR |
| | | | | 0970 | | : | -DESELECT ALL I/O DEVICES |
| | | | | 0980 | RESET | : | |
| CO2D- | 08 | | | 0990 | REM | : | PROC "RESET" |
| CO2E- | 48 | | | 1000 | PHP | : | SAVE STATUS |
| CO2F- | A9 | CA | | 1010 | PHA | : | AND DON'T SCRAMBLE A |
| CO31- | 8D | 26 | 03 | 1020 | LDA #L, OUTP | : | RESET LOW BYTE OF OUTPUT VECTOR |
| CO34- | A9 | F1 | | 1030 | STA OUTV | : | TO DEFAULT |
| CO36- | 8D | 27 | 03 | 1040 | LDA #H, OUTP | : | SAME ACTION |
| CO39- | 68 | | | 1050 | STA OUTV+1 | : | FOR HIGH BYTE |
| CO3A- | 28 | | | 1060 | PLA | : | RESTORE A |
| CO3B- | 20 | 33 | F3 | 1070 | PLP | : | RESTORE STATUS |
| CO3E- | 18 | | | 1080 | JSR DESELECT | : | EXEC: "DESELECT" |
| CO3F- | 60 | | | 1090 | CLC | : | SET "NO ERROR" |
| | | | | 1100 | RTS | : | ENDPROC |
| | | | | 1110 | | : | |
| | | | | 1120 | | : | |
| | | | | 1130 | | : | PROCEDURE "SCAN" (CALLED BY KERNAL NOW) |
| | | | | 1140 | | : | -EXECUTE ORIGINAL PROCEDURE "SELECT" |
| | | | | 1150 | | : | -EXECUTE ORIGINAL ROUTINE |
| | | | | 1160 | | : | -IF DEVICE IS 4, INITIALIZE PRINTER |
| | | | | 1170 | | : | AND REMOVE ERROR |
| | | | | 1180 | | : | -IF OTHER DEVICE, BACK TO KERNAL |
| | | | | 1190 | SCAN | : | |
| CO40- | 20 | 50 | F2 | 1200 | REM | : | PROC: "SCAN" |
| CO43- | 08 | | | 1210 | JSR SELECT | : | EXEC: "SELECT" |
| CO44- | 48 | | | 1220 | PHP | : | SAVE STATUS |
| CO45- | A5 | BA | | 1230 | PHA | : | SAVE ERROR CODE IF ANY |
| CO47- | C9 | 04 | | 1240 | LDA *DEVI | : | GET DEVICE WANTED |
| CO49- | F0 | 05 | | 1250 | CMP #\$04 | : | IF NOT "PRINTER 1" |
| CO4B- | 68 | | | 1260 | BEQ CENTR | : | THEN |
| CO4C- | 28 | | | 1270 | PLA | : | RESTORE ERROR CODE |
| CO4D- | 4C | 79 | C0 | 1280 | PLP | : | RESTORE STATUS |
| CO50- | 68 | | | 1290 | JMP ENDSCAN | : | ELSE |
| CO51- | 28 | | | 1300 | PLA | : | PULL A FROM STACK |
| CO52- | 48 | | | 1310 | PLP | : | OMMIT STATUS |
| CO53- | A5 | B9 | | 1320 | PHA | : | PUSH A BACK ON STACK |
| CO55- | C9 | FF | | 1330 | LDA *SECA | : | GET SECONDARY ADDRESS |
| CO57- | D0 | 04 | | 1340 | CMP #\$FF | : | IF SET BY DEFAULT |
| CO59- | A9 | 60 | | 1350 | BNE STOR | : | THEN |
| CO5B- | 85 | B9 | | 1360 | LDA #\$60 | : | REPLACE BY ZERO |
| | | | | 1370 | STA *SECA | : | INTO SECONDARY ADDRESS |
| | | | | 1380 | REM | : | ENDIF |
| CO5D- | 29 | 0F | | 1390 | AND #\$0F | : | DROP OFFSET \$60 |
| CO5F- | 8D | 92 | C1 | 1400 | STA MODE | : | AND STORE COPY |
| CO62- | C9 | 07 | | 1410 | CMP #\$07 | : | IF SEC. ADDRESS 7 |
| CO64- | F0 | 04 | | 1420 | BEQ NOER | : | OR |
| CO66- | C9 | 00 | | | CMP #\$00 | : | SEC. ADDRESS 0 |

| | | | | | | | | |
|-------|----|----|------|---------|-----|----------|---|--------------------------|
| CO68- | DO | 0B | 1430 | | BNE | ERR | : | THEN |
| CO6A- | 20 | 33 | 1440 | NOER | JSR | DESELECT | : | EXEC: "DESELECT" |
| CO6D- | 20 | 15 | 1450 | | JSR | SETCENTR | : | EXEC: "SETCENTR" |
| CO70- | 68 | | 1460 | | PLA | | : | RESTORE A |
| CO71- | 18 | | 1470 | | CLC | | : | SET "NO ERROR" |
| CO72- | 4C | 79 | 1480 | | JMP | ENSCAN | : | ELSE |
| CO75- | 68 | | 1490 | ERR | PLA | | : | REMOVE A FROM STACK |
| CO76- | A9 | 09 | 1500 | | LDA | #\$09 | : | GET "ILLEGAL DEVICE" |
| CO78- | 38 | | 1510 | | SEC | | : | SET "ERROR" |
| | | | 1520 | ENSCAN | REM | | : | ENDIF |
| | | | 1530 | ENDSCAN | REM | | : | ENDIF |
| CO79- | 60 | | 1540 | | RTS | | : | ENDPROC |
| | | | 1550 | | : | | : | |
| | | | 1560 | | : | | : | |
| | | | 1570 | | : | | : | |
| | | | 1580 | | : | | : | |
| | | | 1590 | | : | | : | |
| | | | 1600 | PUTBUS | REM | | : | PROC "PUTBUS" |
| CO7A- | 48 | | 1610 | | PHA | | : | SAVE CHARACTER |
| CO7B- | A9 | 04 | 1620 | | LDA | #\$4 | : | GET MASK FOR HANDSHAKE |
| | | | 1630 | WAIT | REM | | : | REPEAT |
| CO7D- | 2C | 00 | 1640 | | BIT | BUSY | : | CHECK PRINTER STATUS |
| CO80- | DO | FB | 1650 | | BNE | WAIT | : | UNTIL PRINTER NOT BUSY |
| CO82- | 68 | | 1660 | | PLA | | : | REGET CHARACTER |
| CO83- | 8D | 01 | 1670 | | STA | BUS | : | PUT ON CENTRONICS BUS |
| CO86- | 18 | | 1680 | | CLC | | : | SET "NO ERRORS" |
| CO87- | 60 | | 1690 | | RTS | | : | ENDPROC |
| | | | 1700 | | : | | : | |
| | | | 1710 | | : | | : | |
| | | | 1720 | | : | | : | |
| | | | 1730 | | : | | : | |
| | | | 1740 | | : | | : | |
| | | | 1750 | CRLF | REM | | : | PROC "CRLF" |
| CO88- | 48 | | 1760 | | PHA | | : | SAVE CHARACTER |
| CO89- | A5 | B9 | 1770 | | LDA | *SECA | : | SECONDARY ADDRESS |
| CO8B- | 29 | 0F | 1780 | | AND | #\$0F | : | WHITHOUT OFFSET \$60 |
| CO8D- | 8D | 92 | 1790 | | STA | MODE | : | REPRESENTS CHARACTER SET |
| CO90- | A9 | 0A | 1800 | | LDA | #\$0A | : | GET OKI LF WITH CR |
| CO92- | 20 | 7A | 1810 | | JSR | PUTBUS | : | EXEC: "PUTBUS" |
| CO95- | A9 | 00 | 1820 | | LDA | #\$00 | : | RESET COLUMN COUNTER |
| CO97- | 8D | 91 | 1830 | | STA | COLM | : | TO ZERO |
| CO9A- | 2C | 8F | 1840 | | BIT | PGFL | : | IF PAGING ON |
| CO9D- | 10 | 1A | 1850 | | BPL | ENDCR | : | THEN |
| CO9F- | EE | 90 | 1860 | | INC | LINE | : | ADVANCE LINE COUNT |
| COA2- | AD | 90 | 1870 | | LDA | LINE | : | GET RESULT |
| COA5- | C9 | 42 | 1880 | | CMP | #66 | : | IF NEXT PAGE |
| COA7- | DO | 05 | 1890 | | BNE | NORES | : | THEN |
| COA9- | A9 | 00 | 1900 | | LDA | #\$00 | : | PUT ZERO |
| COAB- | 8D | 90 | 1910 | | STA | LINE | : | INTO LINE COUNTER |
| | | | 1920 | NORES | REM | | : | ENDIF |
| COAE- | C9 | 03 | 1930 | | CMP | #\$03 | : | IF IN TOP MARGIN |
| COB0- | 90 | 04 | 1940 | | BCC | PAGE | : | OR |
| COB2- | C9 | 3F | 1950 | | CMP | #63 | : | IN BOTTOM MARGIN |
| COB4- | 90 | 03 | 1960 | | BCC | NOMARG | : | THEN |
| COB6- | 20 | 88 | 1970 | PAGE | JSR | CRLF | : | EXEC: "CRLF" |
| | | | 1980 | NOMARG | REM | | : | ENDIF |
| | | | 1990 | ENDCR | REM | | : | ENDIF |
| COB9- | 68 | | 2000 | | PLA | | : | RESTORE CRLF |
| COBA- | 60 | | 2010 | | RTS | | : | ENDPROC |
| | | | 2020 | | : | | : | |
| | | | 2030 | | : | | : | |
| | | | 2040 | | : | | : | |
| | | | 2050 | | : | | : | |
| | | | 2060 | | : | | : | |
| | | | 2070 | CLEFT | REM | | : | PROC "CLEFT" |
| COBB- | 48 | | 2080 | | PHA | | : | SAVE A |
| COBC- | 8A | | 2090 | | TXA | | : | SAVE X |
| COBD- | 48 | | 2100 | | PHA | | : | |
| COBE- | AE | 91 | 2110 | | LDX | COLM | : | GET COLUMN IN X |
| | | | 2120 | | REM | | : | IF NOT AT LEFT MARGIN |
| COC1- | FO | 0E | 2130 | | BEQ | CLIGN | : | THEN |
| COC3- | CA | | 2140 | | DEX | | : | DECREASE COLUMN |
| COC4- | A9 | 0D | 2150 | | LDA | #\$0D | : | GET CR WHITHOUT LF |

```

COC6- 20 7A C0 2160
COC9- A9 00 2170
COCB- 8D 91 C1 2180
COCE- 20 D5 C0 2190
          2200 CLIGN
COD1- 68 2210
COD2- AA 2220
COD3- 68 2230
COD4- 60 2240
          2250
          2260
          2270
          2280
          2290 ORGC
COD5- EC 91 C1 2300
COD8- FO 06 2310
CODA- 20 26 C1 2320
CDD- 4C D5 C0 2330
COE0- 60 2340 ENDORGC
          2350
          2360
          2370
          2380
          2390
          2400 CTRL
          2410
          2420
          2430
          2440
          2450
          2460
          2470
          2480
          2490
          2500
          2510
          2520
          2530
          2540
          2550
          2560
          2570
          2580
          2590
          2600 ENDCTRL
C106- 20 7A C0 2610
          2620
C109- 60 2630
          2640
          2650
          2660
          2670
          2680 SETLO
C10A- A9 07 2690
C10C- 8D 92 C1 2700
C10F- 60 2710
          2720
          2730
          2740
          2750
          2760 SETUP
C110- A9 00 2770
C112- 8D 92 C1 2780
C115- 60 2790
          2800
          2810
          2820
          2830
          2840
          2850 CDOWN
C116- 48 2860
C117- 8A 2870
C118- 48 2880

```

```

JSR PUTBUS          : EXEC: "PUTBUS"
LDA #00             : PUT ZERO
STA COLM            : INTO COLUMN
JSR ORGC            : EXEC: "ORGC"
REM                 : ENDIF
PLA                 : RESTORE X
TAX                 :
PLA                 : RESTORE A
RTS                 : ENDPROC
:
:PROCEDURE "ORGC"
: -GO TO COLUMN FROM X
:
REM                 :PROC "ORGC"
CPX COLM            : WHILE NOT ON COLUMN X
BEQ ENDORGC         : DO:
JSR BLANK           : EXEC: "BLANK"
JMP ORGC            : ENDWHILE
RTS                 : ENDPROC
:
:PROCEDURE "CTRL"
: -HANDLE CONTROL CHARACTER
:
REM                 :PROC "CTRL"
REM                 : CASE CHARACTER
CMP #08             : WHEN BACKSPACE
BEQ CLEFT           : GO DO CURSOR LEFT
CMP #91             : WHEN CURSOR UP
BEQ SETUP           : GO SET UPPERCASE
CMP #11             : WHEN CURSOR DOWN
BEQ SETLO           : GO SET LOWERCASE
CMP #0A             : WHEN LINE FEED
BEQ CDOWN           : GO DO CURSOR DOWN
CMP #0D             : WHEN CRLF
BEQ CRLF            : GO DO CRLF
CMP #13             : WHEN HOME
BEQ SETPG           : GO SET PAGING ON
CMP #93             : WHEN CLR/HOME
BEQ CLRPG           : GO SET PAGING OFF
CMP #0C             : WHEN FORMFEED
BNE ENDCTRL         : AND
BIT PGFL            : PAGING ON
BMI NXTPG           : GO TO NEXT PAGE
REM                 : OTHERWISE
JSR PUTBUS          : EXEC: "PUTBUS"
REM                 : ENDCASE
RTS                 : ENDPROC
:
:PROCEDURE "SETLO"
: -SET LOWERCASE/UPPERCASE MODE
:
REM                 :PROC "SETLO"
LDA #07             : SET LOWERCASE
STA MODE            : INTO CHAR. MODE
RTS                 : ENDPROC
:
:PROCEDURE "SETUP"
: -SET UPPERCASE/GRAPHICS MODE
:
REM                 :PROC "SETUP"
LDA #00             : SET UPPERCASE
STA MODE            : INTO CHAR. MODE
RTS                 : ENDPROC
:
:PROCEDURE "CDOWN"
: -PERFORM LINEFEED WITHOUT CR
:
REM                 :PROC "CDOWN"
PHA                 : SAVE A
TXA                 : SAVE X
PHA                 :

```

| | | | | | |
|-------|----------|------|--------------------------------------|-----|---------------------------|
| C119- | AE 91 C1 | 2890 | LDX COLM | : | GET COLUMN IN X |
| C11C- | 20 88 C0 | 2900 | JSR CRLF | : | EXEC: "CRLF" |
| C11F- | 20 D5 C0 | 2910 | JSR ORGC | : | EXEC: "ORGC" |
| C122- | 68 | 2920 | PLA | : | RESTORE X |
| C123- | AA | 2930 | TAX | : | |
| C124- | 68 | 2940 | PLA | : | RESTORE A |
| C125- | 60 | 2950 | RTS | : | ENDPROC |
| | | 2960 | : | | |
| | | 2970 | :PROCEDURE "BLANK" | | |
| | | 2980 | : -PRINT A BLANK | | |
| | | 2990 | : | | |
| | | 3000 | BLANK | REM | :PROC "BLANK" |
| C126- | A9 20 | 3010 | LDA #20 | : | GET A BLANK |
| C128- | 20 2C C1 | 3020 | JSR OUTPUT | : | EXEC: "OUTPUT" |
| C12B- | 60 | 3030 | RTS | : | ENDPROC |
| | | 3040 | : | | |
| | | 3050 | :PROCEDURE "OUTPUT" | | |
| | | 3060 | : -PRINT CHARACTER. COUNT COLUMN | | |
| | | 3070 | : | | |
| | | 3080 | OUTPUT | REM | :PROC "OUTPUT" |
| C12C- | 20 7A C0 | 3090 | JSR PUTBUS | : | EXEC: "PUTBUS" |
| C12F- | EE 91 C1 | 3100 | INC COLM | : | ADVANCE COLUMN |
| C132- | 60 | 3110 | RTS | : | ENDPROC |
| | | 3120 | : | | |
| | | 3130 | :PROCEDURE "CLRPG" | | |
| | | 3140 | : -SWITCH PAGING OFF | | |
| | | 3150 | : | | |
| | | 3160 | CLRPG | REM | :PROC "CLRPG" |
| C133- | 4E 8F C1 | 3170 | LSR PGFL | : | PAGING OFF |
| C136- | 60 | 3180 | RTS | : | ENDPROC |
| | | 3190 | : | | |
| | | 3200 | :PROCEDURE "SETPG" | | |
| | | 3210 | : -SWITCH PAGING ON | | |
| | | 3220 | : | | |
| | | 3230 | SETPG | REM | :PROC "SETPG" |
| C137- | 38 | 3240 | SEC | : | PAGING ON |
| C138- | 6E 8F C1 | 3250 | ROR PGFL | : | |
| C13B- | A9 41 | 3260 | LDA #65 | : | SET "END OF PAGE" |
| C13D- | 8D 90 C1 | 3270 | STA LINE | : | INTO LINE COUNT |
| C140- | A9 00 | 3280 | LDA #00 | : | PUT ZERO |
| C142- | 8D 91 C1 | 3290 | STA COLM | : | INTO COLUMN COUNT |
| C145- | 60 | 3300 | RTS | : | ENDPROC |
| | | 3310 | : | | |
| | | 3320 | : | | |
| | | 3330 | :PROCEDURE "NXTPG" | | |
| | | 3340 | : -GO TO NEXT PAGE | | |
| | | 3350 | : | | |
| | | 3360 | NXTPG | REM | :PROC "NXTPG" |
| C146- | AD 90 C1 | 3370 | LDA LINE | : | WHILE CURRENT LINE NUMBER |
| C149- | C9 03 | 3380 | CMP #03 | : | NOT 3. |
| C14B- | F0 06 | 3390 | BEQ PAGED | : | DO: |
| C14D- | 20 88 C0 | 3400 | JSR CRLF | : | EXEC: "CRLF" |
| C150- | 4C 46 C1 | 3410 | JMP NXTPG | : | ENDWHILE |
| C153- | 60 | 3420 | RTS | : | ENDPROC |
| | | 3430 | : | | |
| | | 3440 | :PROCEDURE "UPLOW" | | |
| | | 3450 | : -PRINT IN UPPERCASE/LOWERCASE MODE | | |
| | | 3460 | : | | |
| | | 3470 | UPLOW | REM | :PROC "UPLOW" |
| C154- | 29 FF | 3480 | AND #FF | : | IF SHIFTED CHARACTER |
| C156- | 10 04 | 3490 | BPL NOSHFT | : | THEN |
| C158- | 29 7F | 3500 | AND #7F | : | DROP BIT 7 |
| C15A- | D0 0B | 3510 | BNE ENDUPL | : | ELSE |
| C15C- | C9 40 | 3520 | CMP #40 | : | IF ABOVE ' ? |
| C15E- | 90 07 | 3530 | BCC NTLW | : | AND |
| C160- | C9 5B | 3540 | CMP #5B | : | BELOW ^ |
| C162- | 80 03 | 3550 | BCS NTLW | : | THEN |
| C164- | 18 | 3560 | CLC | : | ADD OFFSET |
| C165- | 69 20 | 3570 | ADC #20 | : | TO LOWERCASE |
| | | 3580 | REM | : | ENDIF |
| | | 3590 | ENDUPL | REM | ENDIF |
| C167- | 20 2C C1 | 3600 | JSR OUTPUT | : | EXEC: "OUTPUT" |
| C16A- | 60 | 3610 | RTS | : | ENDPROC |

```

3620
3630
3640 : PROCEDURE "PROUT"
3650 : -CHECK CHARACTER TYPE
3660 : -CHECK CHARACTER SET TYPE
3670 : -HANDLE CHARACTER ACCORDINGLY
3680
3690 PROUT REM : PROC "PROUT"
3700 PHA : SAVE CHARACTER
3710 AND ##7F : DROP SHIFT BIT
3720 CMP ##20 : IF CTRL CHARACTER
3730 BCS NORMAL : THEN
3740 PLA : REGET FROM STACK
3750 PHA : SAVE CHARACTER
3760 JSR CTRL : EXEC: "CTRL"
3770 JMP ENDPR1 : ELSE
3780 NORMAL LDA MODE : IF LOWERCASE MODE
3790 BEQ UPGR : THEN
3800 PLA : REGET CHARACTER
3810 PHA : SAVE CHARACTER
3820 JSR UPLOW : EXEC: "UPLOW"
3830 JMP ENDPR2 : ELSE
3840 UPGR PLA : REGET CHARACTER
3850 PHA : SAVE CHARACTER
3860 JSR OUTPUT : EXEC: "OUTPUT"
3870 ENDPR2 : ENDIF
3880 ENDPR1 : ENDIF
3890 PLA : RESTORE CHARACTER
3900 CLC : SET "NO ERRORS"
3910 RTS : ENDPROC
3920
3930 : STORAGE AREA
3940
3950 PGFL .BY $00 : PAGING ON (BIT 7)
3960 LINE .BY $00 : CURRENT LINE NUMBER
3970 COLM .BY $00 : CURRENT COLUMN
3980 MODE .BY $00 : TYPE CHARACTER SET
3990
//0000,C193,C193

```

Brief aan de redactie.

Bert van Tiel, Gouda.

Ik werk nu sinds een half jaar met de JUNIOR + interface. 3 x 16 K dynamische ram, VDU-kaart en diskcontroller met daaraan een 80-tracks drive. merk Micropolis. Ik heb daardoor uitwisselproblemen gehad met de 40-tracks drives van andere leden, die me dan ook aanraadden dit softwarematig op te lossen; dus steeds 2 tracks in plaats van 1 track springen. Dit heeft me echter op een ander soort gebracht, namelijk de DOS van OHIO aanpassen op 80 tracks. Welnu, dat wilde ik je hierbij doorgeven. De volgende veranderingen zijn daarvoor nodig:

26C9 C9 40 wordt C9 80 : aanpassen ERROR-
melding voor 40 tracks
2768 A9 39 wordt A9 79
2778 C9 39 wordt C9 79
2DA6 A9 39 wordt A9 79

Misschien dat andere leden met een of meerdere 80-tracks drives hiermee hun voordeel kunnen doen.

VRAAG EN AANBOD

FATE 65 IS A 12K (F)ORMAT LISTER. (A)SSEMBLER. (T)APE-UTILITIES. (E)DITOR FOR JUNIOR-COMPUTER WITH PRINTER-MONITOR AND TAPE-MONITOR. THE SOURCE-LISTING WAS DEVELOPED BY ROB BANEN: COPYRIGHTS OF ASSEMBLER AND EDITOR BY PROTON ELECTRONICS, NAARDEN, THE NETHERLANDS. WE CAN OFFER YOU THIS USEFULL PROGRAM WITH THE PERMISSION OF PROTON ELECTRONICS.

WITH THE EDITOR ONE CAN MAKE TEXT-FILES AND SOURCE-FILES. TO BE WRITTEN ON OR READ FROM ONE OR TWO TAPE-RECORDERS. WITH OR WITHOUT REMOTE-CONTROL. WITH THE ASSEMBLER IT IS POSSIBLE TO CONVERT SOURCE-FILES INTO OBJECT-CODE. FROM CASSETTE TO CASSETTE. FROM MEMORY TO MEMORY. FROM CASSETTE TO MEMORY AND VISA VERSA. YOU CAN MERGE AND YOU CAN SPLIT. THE FORMAT-LISTER IS MADE TO CONTROL THE MICROLINE PRINTER OKI-80. BUT IT IS EASY TO CHANGE FOR YOUR OWN PRINTER. YOU NEED 16K FREE RAM. BETTER 32K (\$2000 - \$9000). CASSETTE WITH OBJECT IN KIM/JUNIOR HYPERTAPE AND MANUAL (DUTCH LANGUAGE) FOR FATE 65 V1.0 FL. 47.50
COMPLETE SOURCE-LISTING WITH COMMENTS
(ENGLISH LANGUAGE) FOR FATE 65 V1.0 FL. 110.==
MEMBERS IN FOREIGN COUNTRIES PAY WITH EUROCHEQUES.
OTHERWISE HFL. 7.50 EXTRA TRANSFERS.

COMMODORE BASIC 2.0 (called Upgrade or New ROMs (For CBM 30XX)).

Tokenized Microsoft Basic Keywords and addresses N. de Vries
Analogous to the publication for Commodore-64 by A. Mueller.
DE 6502 KENNER, December 1983, pages 5-8.

| COMMANDS | | | | | |
|---|-------|--------|--------|----|--------|
| KEYWORDS CORRESPONDING TO C092 | | | | | |
| ADDRESSES CORRESPONDING TO C000 | | | | | |
| THE ADDRESSES OF ROUTINES FOR COMMANDS ARE THE ADDRESSES MINUS 1. BECAUSE THE ROUTINES ARE INVOKED THROUGH RTS. | | | | | |
| KEYWORD | TOKEN | ADDR-1 | | | |
| END | 80 | C740 | WAIT | 92 | D707 |
| FOR | 81 | C657 | LOAD | 93 | FFD4 |
| NEXT | 82 | CC1F | SAVE | 94 | FFD7 |
| DATA | 83 | C7FF | VERIFY | 95 | FFDA |
| INPUT# | 84 | CAA6 | DEF | 96 | D28C |
| INPUT | 85 | CAC0 | POKE | 97 | D706 |
| DIM | 86 | CF62 | PRINT# | 98 | C98A |
| READ | 87 | CB06 | PRINT | 99 | C9AA |
| LET | 88 | C8AC | CONT | 9A | C76A |
| GOTO | 89 | C7AC | LIST | 9B | C5B4 |
| RUN | 8A | C784 | CLR | 9C | C576 |
| IF | 8B | C82F | CMD | 9D | C990 |
| RESTORE | 8C | C72F | SYS | 9E | FFDE |
| GOSUB | 8D | C78F | OPEN | 9F | FFBF |
| RETURN | 8E | C7D9 | CLOSE | A0 | FFC2 |
| REM | 8F | C842 | GET | A1 | CA7C |
| STOP | 90 | C73E | NEW | A2 | C55A |
| ON | 91 | C852 | GO | CB | (C725) |
| NULL | - | - | | | |

| MISCELLANEOUS KEYWORDS | |
|--------------------------------|-------|
| KEYWORDS CORRESPONDING TO C11D | |
| KEYWORD | TOKEN |
| TAB(| A3 |
| TO | A4 |
| FN | A5 |
| SPC(| A6 |
| THEN | A7 |
| NOT | A8 |
| STEP | A9 |

COMMODORE BASIC 2.0 (called Upgrade or New ROMs (For CBM 30XX)).

Tokenized Microsoft Basic keywords and addresses

DYADIC OPERATORS

KEYWORDS CORRESPONDING TO C134

PRIORITIES AND ADDRESSES CORRESPONDING TO C074

THE ADDRESSES OF ROUTINES FOR DYADIC OPERATORS ARE THE ADDRESS MINUS 1. BECAUSE THE ROUTINES ARE INVOKED THROUGH A RTS INSTRUCTION.

| KEYWORD | TOKEN | ADDR-1 | PRTY |
|---------|-------|--------|------|
|---------|-------|--------|------|

| | | | | |
|-------------|----|------|----|----------------|
| + | AA | D775 | 79 | addition |
| - | AB | D735 | 79 | subtraction |
| * | AC | D936 | 7B | multiplication |
| / | AD | DA1D | 7B | division |
| ^ | AE | DE67 | 7F | exponentiation |
| AND | AF | CECA | 50 | logical AND |
| OR | BO | CEC7 | 46 | logical OR |
| monadic "-" | AB | DEA0 | 7D | negation |
| monadic NOT | A8 | CDCE | 5A | logical NOT |
| > | B1 | CEF7 | 64 | comparison |
| = | B2 | CEF7 | 64 | comparison |
| < | B3 | CEF7 | 64 | comparison |

FUNCTIONS

FUNCTIONS CORRESPONDING TO C141

ADDRESSES CORRESPONDING TO C046

| KEYWORD | TOKEN | ADDR |
|---------|-------|------|
|---------|-------|------|

| | | | | | |
|-----|----|------|---------|----|------|
| SGN | B4 | DB45 | TAN | C0 | E028 |
| INT | B5 | DBD8 | ATN | C1 | E08C |
| ABS | B6 | DB64 | PEEK | C2 | D6E8 |
| USR | B7 | 0000 | LEN | C3 | D656 |
| FRE | B8 | D259 | STR\$ | C4 | D33F |
| POS | B9 | D27A | VAL | C5 | D687 |
| SQR | BA | DE5E | ASC | C6 | D665 |
| RND | BB | DF7F | CHR\$ | C7 | D5C6 |
| LOG | BC | D8F6 | LEFT\$ | C8 | D5D5 |
| EXP | BD | DEDA | RIGHT\$ | C9 | D606 |
| COS | BE | DFD8 | MID\$ | CA | D611 |
| SIN | BF | DFDF | | | |

- Note: 1) Bit 7 in the last character of each keyword is set to determine the end of a keyword
 2) Keywords are tokenized by adding x'80' to their relative (hex) position in the table.

FORTH OP JUNIOR COMPUTERS DEEL 2.

door: G. van Opbroek (GEVOP)
Hooglanden 28
9801 LB Zuidhorn

1. Inleiding.

In deze aflevering zal ik trachten enige duidelijkheid te verschaffen in het gebruik van de verschillende versies van FORTH voor de Junior en hoe programma's die in de ene versie geschreven zijn omgewerkt kunnen worden naar een andere versie.

Voor de Junior zijn (bij mijn weten) de volgende versies beschikbaar:

- 1) fig-FORTH
- 2) 79-standaard FORTH voor gebruik met cassette
- 3) 79-standaard FORTH voor de DOS van Koen van Nieuwenhove
- 4) 79-standaard FORTH voor de Senior
- 5) PE-FORTH voor de Senior

Hoewel versie 2, 3 en 4 allen de naam 79-standaard dragen voldoet niet een van deze FORTH versies in alle opzichten aan de 79-standaard.

2. De 79-standaard voor FORTH.

FORTH dat aan de 79-standaard voldoet bevat de volgende woorden [1]:

Stack manipulatie:

| | | | | | | |
|------|------|------|------|-------|------|------|
| DUP | DROP | SWAP | OVER | ROT | PICK | ROLL |
| ?DUP | >R | R> | R0 | DEPTH | | |

Vergelijking:

| | | | | | | |
|----|-----|---|----|----|----|----|
| < | = | > | 0< | 0= | 0> | D< |
| UK | NOT | | | | | |

Rekenkundig en logisch:

| | | | | | | |
|-------|-----|-----|------|--------|---------|-----|
| + | D+ | - | 1+ | 1- | 2+ | 2- |
| % | / | MOD | /MOD | %/MOD | %/ | UX |
| U/MOD | MAX | MIN | ABS | NEGATE | DNEGATE | AND |
| OR | XOR | | | | | |

Geheugen:

| | | | | | | |
|-------|------|----|----|---|---|------|
| @ | ! | CE | C! | ? | + | MOVE |
| CMOVE | FILL | | | | | |

Besturings structuren:

```
DO .... LOOP
I      J      LEAVE
DO .... +LOOP
IF .... THEN
IF .... ELSE .. THEN
BEGIN ..... UNTIL
BEGIN ..... WHILE ..... REPEAT
EXIT      EXECUTE
```

Terminal invoer/uitvoer:

| | | | | | | |
|-----|--------|-------|--------|------|-------|-----------|
| CR | EMIT | SPACE | SPACES | TYPE | COUNT | -TRAILING |
| KEY | EXPECT | QUERY | WORD | | | |

Nummerieke conversie:

| | | | | | |
|------|---------|------|---------|----|---|
| BASE | DECIMAL | U. | CONVERT | <# | # |
| #S | HOLD | SIGN | #) | | |

Massa geheugen invoer/uitvoer:

| | | | | | |
|--------------|------|---------------|-------|--------|--------|
| LIST | LOAD | SCR | BLOCK | UPDATE | BUFFER |
| SAVE-BUFFERS | | EMPTY-BUFFERS | | | |

Definierende woorden:

| | | | |
|------------|--------|----------|----------|
| : | : | VARIABLE | CONSTANT |
| VOCABULARY | CREATE | | DOES) |

Woordenlijsten:

| | | | | |
|---------|---------|-------|-------------|------|
| CONTEXT | CURRENT | FORTH | DEFINITIONS | FIND |
| FORGET | | | | |

Vertaler:

| | | | | |
|-------|-------|---|-----------|-----------|
| : | ALLOT | ; | IMMEDIATE | LITERAL |
| STATE | [|] | COMPILE | [COMPILE] |

Diversen:

| | | | | | | |
|-------------|------|-----|-----|-----|-------|------|
| < | HERE | PAD | >IN | BLK | ABORT | QUIT |
| 79-STANDARD | | | | | | |

Verder schrijft de standaard voor dat het massa geheugen opgedeeld is in blokken met een lengte van 1024 bytes.

Behalve het bovengenoemde standaardset kent 79-FORTH nog een aantal extra woorden. Deze woorden hoeven niet in een FORTH volgens de 79-standaard aanwezig te zijn, maar zijn in de voor de JUNIOR beschikbare 79-standaard FORTH meestal wel aanwezig. Dit zijn:

Double number uitbreiding:

| | | | | | |
|------|---------|-----------|-------|------|-------|
| 2! | 20 | 2CONSTANT | 2DROP | 2DUP | 2OVER |
| 2ROT | 2SWAP | 2VARIABLE | D+ | D- | D. |
| D.R | D0= | D< | D= | DABS | DMAX |
| DMIN | DNEGATE | DUK | | | |

Reference woord set:

| | | | | | |
|-------|--------|-----------|--------|--------|-----|
| --> | -MATCH | -TEXT | .R | 2X | 2/ |
| IS | AGAIN | B/BUF | BL | BLANKS | C. |
| FLUSH | HEX | INTERPRET | NUMBER | S0 | SP0 |
| TEXT | VLIST | | | | |

Bovendien bevat FORTH meestal een EDITOR en een ASSEMBLER.

3. fig-FORTH.

fig-FORTH wordt verspreid door de Forth Interest Group en is als source listing beschikbaar voor vrijwel alle gangbare microprocessors. Dit is ook het geval voor de door ons gebruikte 6502. De source listing van fig-FORTH is beschikbaar [2] en is door Gert Klein [3] bewerkt voor Junior onder K&N-DOS en door mij voor de Senior [4]. Bovendien is de versie voor een Junior zonder schijven die door Gerard van Roekel beschreven is [5] afgeleid van [2] en uitgebreid met cassette routines.

fig-FORTH wijkt in een aantal opzichten enigzins af van

de 79-standaard. In de eerste plaats ontbreken de woorden:

| | | | | | | |
|------|------|-------|------|----|---------|------|
| PICK | ROLL | DEPTH | D= | 0> | DC | UC |
| 1- | 2- | J | EXIT | U. | CURRENT | FIND |
| WORD | | | | | | |

Verder zijn er enkele woorden die een andere naam hebben:

| | | | |
|---------------|--------|---------------|----------|
| 79-standaard: | fig: | 79-standaard: | fig: |
| ?DUP | -DUP | NOT | 0= |
| U/MOD | U/ | NEGATE | MINUS |
| DNEGATE | DMINUS | CONVERT | (NUMBER) |
| SAVE-BUFFERS | FLUSH | >IN | IN |
| RE | R | | |

Bovendien kent fig-FORTH een aantal woorden waar geen equivalent in de 79-standaard voor is. Daar echter een 79-standaard FORTH meestal met behulp van conversies uit een fig-FORTH gebouwd wordt, zijn deze woorden over het algemeen wel in een 79-standaard FORTH aanwezig.

Tenslotte zijn er nog enkele afwijkingen in het gebruik van de woorden in fig-FORTH:

- 1) Variabelen worden bij de definitie geïnitialiseerd:
n VARIABLE xxxx
- 2) De combinatie CREATE DOES uit de standaard bestaat niet. In de plaats hiervan kan de combinatie:
<BUILDS DOES>

gebruikt worden. Het woord CREATE in fig-FORTH heeft een andere betekenis als CREATE in de 79-standaard. In fig-FORTH wordt CREATE meestal alleen door FORTH zelf gebruikt.

- 3) Bij de constructie:

n1 n2 DO n3 +LOOP

wordt bij een negatieve n3 gestopt als de index gelijk wordt aan of kleiner wordt dan n1. Bij 79-standaard FORTH wordt er gestopt als de index kleiner wordt dan n1.

- 4) fig-FORTH werkt niet goed als er gewerkt wordt met lege tekststrings, dit in tegenstelling tot 79-standaard FORTH.

4. 79-standaard FORTH op JUNIOR.

De zogenaamde 79-standaard versies van FORTH die voor de JUNIOR beschikbaar zijn wijken alle in meerdere of mindere mate van deze standaard af. Voor alle versies geldt namelijk dat de block-lengte geen 1024 bytes is zoals deze standaard voorschrijft. In de meeste gevallen bedraagt de block-lengte 128 bytes, in een enkel geval (double-density DOS) 256 bytes. De block-lengte van een FORTH is op te vragen met behulp van het woord "B/BUF".

In de FORTH voor de JUNIOR met cassettes (en de FORTH voor KVN DOS ?) geldt bovendien dat het woord DOES niet aan de standaard voldoet. Dit heeft tot gevolg dat de combinatie "CREATE DOES" niet werkt en vervangen moet worden door de combinatie "<BUILDS DOES". Met behulp van een assembler in FORTH is dit probleem op eenvoudige wijze op te lossen. Ik ben van plan op deze zaak nog terug te komen als er voor de betreffende FORTH-versies een assembler beschikbaar is.

Tenslotte ontbreken vaak de woorden "S0", "DEPTH" en "WORD".

5. PE-FORTH voor de SENIOR.

Behalve de van fig-FORTH afgeleide versies bestaat er nog een FORTH die speciaal bedoeld is voor SENIOR DOS. Deze FORTH wijkt sterk af van de andere beschreven FORTH versies en omzetting van 79-FORTH naar PE-FORTH of anderszins zal enkele problemen kunnen geven. Het belangrijkste verschil van PE-FORTH ten opzichte van de 79-standaard is de aanwezigheid van zgn. floating-point routines zodat het in PE-FORTH mogelijk is met drijvende komma getallen te werken. Een tweede verschil is het gebruik van variabelen. PE-FORTH volgt het zgn. "TO" concept wat inhoudt dat het noemen van de naam van een variabele de waarde van de variabele op de stack zet i.p.v. het adres van de variabele. Het geven van een waarde aan een variabele gaat met de constructie:

waarde TO naam i.p.v. met waarde naam !

Evenals bij fig-FORTH wordt bij de definitie van een variabele de startwaarde van deze variabele meegegeven en wordt ook de constructie "<BUILDS DOES>" gebruikt.

Wat betreft de woorden die in PE-FORTH aanwezig zijn lijkt deze versie meer op fig-FORTH dan op de 79-standaard. Opvallend is echter dat alle woorden die op losse bytes betrekking hebben beginnen met een "B" i.p.v. een "C":

BE, B!, BMOVE i.p.v. CE, C!, CMOVE

Tenslotte kent PE-FORTH nog een groot aantal uitbreidingen. Dit zijn onder andere de woorden:

CASE HOME BELL H. B. BLOCKS -UPDATE
ORPHAN :ORPHAN ADOPT MYSELF FORWARD ONERR> EXTERNAL

Bovendien kent PE-FORTH het woord "\" dat betekent: de rest van de regel is commentaar en hoeft niet geladen te worden.

6. Afsluiting.

In de vorige paragrafen heb ik trachten aan te geven wat de verschillen tussen de verschillende FORTH versies zijn. Hoewel in enkele gevallen de verschillen aanzienlijk zijn is het toch gebleken dat programma's die in de ene versie geschreven zijn met een beetje inventiviteit goed zijn om te werken naar een andere versie.

Hoewel de beschikbare FORTH versies steeds vergeleken zijn met de 79-standaard wil ik toch nog even opmerken dat er sinds kort een nieuwe standaard voor FORTH bestaat. Dit is de 83-standaard en is beschreven in [6].

Misschien komt er in de toekomst een 83-standaard versie voor de JUNIOR beschikbaar of een versie van de zeer krachtige FYS-FORTH die momenteel uitsluitend nog op APPLE draait.

7. Literatuur.

- 1) FORTH-79 STANDARD
- 2) fig-FORTH 6502 ASSEMBLY SOURCE LISTING
Release 1.1
FORTH INTEREST GROUP
- 3) G. Klein
fig-FORTH SOURCE
- 4) G. van Opbroek
fig-FORTH SOURCE for SENIOR
- 5) G. van Roekel
Using FORTH with the 6502,
De 6502 Kenner 28 pag. 15,
Oktober 1983.
- 6) C. Kevin McCabe
FORTH-83 EVOLUTION CONTINUES,
Byte augustus 1984 pag. 137.

0 6 MLIST

SCR # 0

```

0 ( FORTH-EXTENSIONS )
1 ( door : Fridus Jonkman , Stijn Streuvelsstraat 9 , )
2 ( 5242 GD Rosmalen , tel. 04192-16146 )
3 ( ===== )
4 ( Deze screens bevatten een aantal FORTH-woorden, die voor )
5 ( mij een nuttige uitbreiding vormen. Ik werk met een fig- )
6 ( FORTH op een uitgebreide Junior met VDU-kaart en nog cas- )
7 ( settes als opslagmedium. Fig-FORTH begint bij mij op adres )
8 ( $2000. )
9
10 ( FORTH en ASSEMBLER ----- )
11 ( Hoewel er een goede 6502-assembler in FORTH is van W. Ragsda- )
12 ( le, had ik toch behoefte aan een woord, waarmee je van FORTH )
13 ( naar een subroutine in machinetaal kunt springen en ook weer )
14 ( terug kunt keren naar FORTH, zonder een assembler te gebrui- )
15 ( ken. JSR is zo'n woord. )

```

SCR # 1

```

0 ( JSR (( jumpadres --- )) )
1 FORTH DEFINITIONS HEX
2 6 USER S0 ( S0 bevat initiele stackpointer $009E )
3 : JSR HERE ! ( jumpadres naar HERE )
4 SP@ S0 C! ( save huidige stackpointer in S0 )
5 HERE ( indirect jumpadres naar stack )
6 ' LIT 17 + > R ( returnadres-1 = NEXT naar returnstack )
7 EXECUTE ( voer machinetaalroutine uit en return )
8 SP! ( haal stackpointer terug uit S0 )
9 9E S0 C! ( laad S0 met initiele waarde )
10
11 ( De volgende woorden maken cursorbesturing mogelijk; er is )
12 ( gebruik gemaakt van machinetaalroutines uit de Elektuur- )
13 ( software bij de VDU-kaart. De jumpadressen zijn hier in )
14 ( constanten ondergebracht; pas ze aan je eigen systeem aan. )
15 --)

```

SCR # 2

```

0 ( Cursorbesturing )
1 1512 CONSTANT CLNUP 1551 CONSTANT CLNDN 15CE CONSTANT ADJCUR
2 1A38 CONSTANT INLINE 1A39 CONSTANT COL
3 ( !CURSOR (( X Y --- )) ; plaats cursor naar X.Y )
4 : !CURSOR INLINE C@ > R DUP ROT COL C! INLINE C!
5 R) SWAP --DUP IF 0 OVER 0( 0=
6 IF DO CLNUP JSR LOOP
7 ELSE SWAP DO CLNDN JSR LOOP
8 THEN
9 THEN ADJCUR JSR :
10 ( ?CURSOR (( --- X Y )) ; huidige cursorpositie naar stack )
11 : ?CURSOR COL C@ INLINE C@ ;
12 ( +CURSOR (( +X +Y --- )) ; verplaats cursor t.o.v. huidige )
13 ( positie met +X,+Y )
14 : +CURSOR ?CURSOR ROT + > R SWAP + R) !CURSOR ;
15 --)

```

SCR # 3

```

0 ( Omschakelen van normale naar grafische mode en v.v. )
1 ( via de woorden NORMAL en GRAPHIC . De gegeven waarden zijn )
2 ( enigszins afhankelijk van je monitor en van het kristal in )
3 ( de VDU-kaart. Ik gebruik een 15 Mhz kristal. )
4 ( Pas de waarden in de constanten weer aan aan je systeem. )
5 E800 CONSTANT AR E801 CONSTANT RFILE
6 : !AR AR C! ; ( !AR (( registernr. 6845 --- )) )
7 : !RFILE RFILE C! ; ( !RFILE (( waarde --- )) )
8 : NORMAL 4 !AR 21 !RFILE 5 !AR 6 !RFILE 7 !AR 1D !RFILE
9 9 !AR 8 !RFILE A !AR 68 !RFILE B !AR 8 !RFILE ;
10 : GRAPHIC 4 !AR 26 !RFILE 5 !AR 0 !RFILE 7 !AR 1F !RFILE
11 9 !AR 7 !RFILE A !AR 67 !RFILE B !AR 7 !RFILE ;
12
13
14
15 --)

```

```

SCR # 4
0 ( Grafische woorden voor hor. en vert. lijn en voor rechthoek )
1 ( Het character-byte dient om de dikte van de lijn te bepalen: )
2 ( zie voor de waarden de karaktergenerator. )
3 ( HLINE (( char.byte lengte X Y --- )) )
4 : HLINE !CURSOR 0 DO DUP EMIT LOOP DROP ;
5 ( VLINE (( char.byte hoogte X Y --- )) )
6 : VLINE GRAPHIC !CURSOR 0 DO DUP EMIT -1 1 +CURSOR LOOP DROP ;
7 ( RECTANGLE (( lengte hoogte X Y --- )) )
8 : RECTANGLE !CURSOR >R >R ?CURSOR R) R) CC EMIT OVER 83 SWAP
9 ?CURSOR HLINE CD EMIT -1 1 +CURSOR 8B OVER ?CURSOR
10 VLINE CE EMIT >R >R 1+ OVER OVER !CURSOR R) R) 8C
11 SWAP ?CURSOR VLINE CB EMIT 84 SWAP ?CURSOR HLINE
12 SWAP 1+ SWAP !CURSOR ;
13
14
15 --)

```

```

SCR # 5
0 ( Nieuwe ID. en MON, ECHO-ON en ECHO-OFF. GO )
1 ( Door FORTH wordt het meest significante bit van een byte )
2 ( niet ge"and" met $7F, via ID. gaat de laatste letter van elk )
3 ( FORTH-woord de mist in. Dit is te verhelpen door ID. te )
4 ( herdefinieren. Let er wel op dat daarna ook elk woord, dat )
5 ( gebruik maakte van de oude ID., ook opnieuw na ID. moet wor- )
6 ( den gecompileerd, zoals VLIST en CREATE. )
7 : ID. PAD 20 5F FILL DUP PFA LFA OVER - PAD SWAP OVER OVER + >R
8 CMOVE R) 1 - DUP C@ 7F AND SWAP C! PAD COUNT 1F AND
9 TYPE SPACE ;
10 ( De volgende woorden spreken voor zich en zijn handig: GO )
11 ( voor het opstarten en MON voor het verlaten van FORTH. )
12 1A41 CONSTANT ECHO : ECHO-OFF 0 ECHO C! : ECHO-ON 1 ECHO C! :
13 : GO BREAK EMPTY-BUFFERS ECHO-OFF 0 DISC ;
14 : MON ECHO-ON MON ;
15 --)

```

```

SCR # 6
0 ( CHECK (( scr#start scr#end --- )) : dit woord checkt )
1 ( screens op aanwezige non-printable karakters, die moeilijk )
2 ( op te sporen fouten bij het "loaden" kunnen veroorzaken. )
3 : CHECK 1+ BORDER SWAP BORDER CR
4 DO HEX 1 C@ 20 (
5 IF 1 4 .R SPACE 1 C@ 2 .R SPACE 1 6000 - 400 /MOD
6 DECIMAL ." SCR# " 2 .R SPACE 40 /MOD ." LINE# "
7 2 .R SPACE ." CHAR# " 2 .R CR
8 THEN
9 LOOP ;
10 ;S
11
12
13
14
15

```

6502 MACHINETAAL

OK

Op het gebied van machinetaal voor de 6502 proces-
sor (de 6510 van de Commodore 64 werkt hier even-
eens mee) verschijnen de laatste tijd wat meer in-
formaties. In de vorige editie hebben wij al gewe-
zen op het bestaan van het Zakboek 6502 van auteur
Bob Bright dat voor Fl. 17,50 te verkrijgen is en
het bestelnummer ISBN 906449 0287 heeft. Nu heeft
Academic Service te Den Haag het boek Machinetaal
voor de 6502 het licht doen zien, ideaal voor de
de beginners op dit gebied. Het boek werd geschre-
ven door A.P. Stephenson, en is nu vanuit het En-
gels vertaald. De prijs bedraagt Fl. 37,50 voor
maar liefst 200 pagina's. Het bestelnummer is ISBN
90.6233.123.8.

BLOKKEREN VAN HET ASCII-KEYBOARD

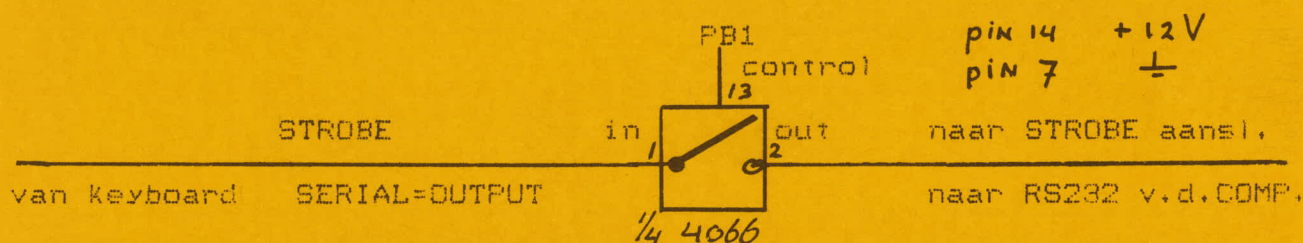
Door: A.C. Tijmons te Rotterdam

voor de Elektuur JUNIOR-computer

Het komt nogal eens voor dat bij het uitvoeren van een Basic-programma per ongeluk of te snel een toets wordt ingedrukt op het keyboard, waardoor het programma onderbroken wordt en in het ergste geval de computer zich terugmeldt met JUNIOR.

Veel toetswerk kan dan verloren zijn gegaan in bijvoorbeeld een bestands-programma, omdat het programma opnieuw gestart moet worden, waarbij alle variabelen gereset worden.

Een oplossing voor dit probleem vormt een klein beetje software en een klein beetje hardware. Er wordt gebruik gemaakt van een elektronische schakelaar die het STROBE-sigitaal, of bij een seriele uitgang het signaal SERIAL-OUTPUT onderbreekt, en enkele PEEK- en POKE-commando's die de elektronische schakelaar besturen met behulp van de PIA van de JUNIOR.



Bij een seriele keyboard aansluiting wordt SERIAL-OUTPUT onderbroken.
Bij een parallelle keyboard aansluiting wordt STROBE onderbroken.

De software:

```
1 GOSUB 230:GOTO1000
219 REM *** SUBROUTINE 220 DEBLOKKEERT ASCII-KEYBOARD
220 POKE 6786,PEEK(6786)+2:RETURN
229 REM *** SUBROUTINE 230 BLOKKEERT ASCII-KEYBOARD
230 POKE 6786,PEEK(6786)-2:RETURN
1000 REM BEGIN
1010 GOSUB 220:INPUT A:GOSUB 230
```

Het adres van PBD is \$1A82=6786.

Na het starten van PM op \$1000 is PB1 uitgang en het niveau "hoog". Subroutine 230 verlaagt \$1A82 met 2, dat wil zeggen: PB1 wordt "laag" en de schakelaar wordt geopend en het keyboard geblokkeerd. Subroutine 220 verhoogt \$1A82 met 2 dus PB1 wordt weer "hoog". Het is van belang dat aan het begin van het programma eerst subroutine 230 wordt doorlopen. Voor en na elk INPUT of GET-commando moeten de beide subroutine's worden doorlopen, hetgeen meer programmeerwerk inhoudt. Bij het aanpassen van een bestaand programma moeten de beide subroutines zo dicht mogelijk in de buurt staan van INPUT of GET.

De schrijver is geïnteresseerd in andere oplossingen van dit probleem die bovengenoemde nadelen niet hebben.

FATE 65

V1.0

